

REMARKS

Claim 4 was previously cancelled. Claims 1-3 and 5-22 are pending in the application.

Rejections of Claims 1-3, 5, 8 and 13-14 Under 35 U.S.C. §103(a)

Applicant respectfully traverses the Examiner's rejection of claims 1-3, 5, 8 and 13-14 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent Application 20002/001096 to Kaminsky (hereinafter referred to as "Kaminsky") in view of U.S. Patent 6,055,513 to Katz (hereinafter referred to as "Katz"). Applicant submits that the newly cited reference of Kaminsky is not available as prior art because its filing date of January 9, 2002 is later than the present Application's effective filing date of October 22, 1999. That is, October 22, 1999 is the filing date of U.S. Patent Application No. 09/426,107 and subject matter from which the instant Application claims priority.

Kaminsky is related to three provisional patent applications (attached). The cited portions of Kaminsky, however, are only contained in one of these three provisionals, that is, provisional application 60/260,583 filed January 9, 2001. This effective filing date of the material relied on by the Examiner is later than the instant Application's effective filing date of October 22, 1999. Thus Kaminsky is not prior art against the instant Application. The cited portions of Kaminsky are NOT contained in provisional application 60/156,020 filed September 23, 1999 and thus the Examiner cannot claim that Kaminsky has an effective date prior to the instant Application. The cited portions of Kaminsky are also NOT contained in provisional application 60/180,466 filed February 4, 2000.

Applicant submits hereinafter with respect to Katz (and the other references cited below) that there are various bases of distinction, and that an essential difference is that Applicant applies rules via a rule processor to two distinct data sets, i.e., "a first data set of user information" and a "second data set of said third party information relating to targeted market segments."

Rejection of Claims 1 and 14:

Claim 1 recites:

“A data mining system comprising:

one or more subscriber servers for collecting information identifying a user and providing a first data set of user information;

one or more demographic databases having third party information relating to targeted market segments and providing a second data set of said third party information relating to targeted market segments; and

a processor in operative communication with the one or more subscriber servers and the one or more demographic databases and receiving said first data set from the one or more subscriber servers and said second data set from the one or more demographic databases,

said processor including a rule processor receiving said first data set and said second data set and applying said first and second data sets to one or more rules to determine a score predicting behavior relating to said collected information identifying said user.”

(Emphasis added; Independent claim 14 recites similar limitations).

The Examiner states that “Kaminsky did not teach [a] second data set is provided by one or more demographic database” (see paragraph 7 of the Office Action) and the Examiner relies on Katz to teach solely such features of claims 1 and 14. Katz, however, fails to disclose or suggest a first and second database such as applicant particularly discloses and claims. Further, Katz does not disclose or suggest applying a first and a second data sets to rules as claimed in the instant application. In contrast, Katz is directed towards a telemarketing system adapted for the selection and upsell of products to a customer based upon the primary transaction data and other information. The Examiner cites col. 8, line 63 to col. 9, line 2; and col. 9, line 65 to col. 10, line 19 of Katz in rejecting Applicant’s claims. This portion of Katz merely states that:

“the system includes primary transaction data and at least a second data element obtained from a database, especially a remote, third party database or databases. Primary transaction data may include data relating to or reflecting the initial or primary contact from the customer to the system...A demographic database may be utilized to identify direct or predicted attributes of the customer...the third party database may provide responsive, effective information for the upsell determination...[see Fig. 2 and also col. 18, line 40-col. 19, line 15; col. 23, lines 6-19] ”

Applicant respectfully submits that the above-quoted portion of Katz does not teach or suggest, “a processor ... receiving said first data set from the one or more subscriber servers and

said second data set from the one as more demographic databases, said processor including a rule processor receiving said first data set and said second data set and applying said first and second data sets to one or more rules to determine a score predicting behavior relating to said collected information identifying said user,” as particularly claimed by Applicant. Again, there is no mention whatsoever of receiving a first data set from one or more subscriber servers and a second data set from one as more demographic databases. There is no mention or suggestion in Katz of determining a score via a rule processor applying rules to a first and second data set.

Specifically, Applicant applies rules via a rule processor to two distinct data sets, i.e., “a first data set of user information” and a “second data set of said third party information relating to targeted market segments.” Since each and every element of independent claim 1 is not present as in the claims, Applicant respectfully submits that claim 1 is patentable over Kaminsky in view of Katz under 35 U.S.C. 103(a). Accordingly, the rejections of independent claims 1 and 14, and claims 2-3, 5, 8 and 13 ultimately depending therefrom are improper and should be withdrawn.

Rejections of Claims 6-7 and 15-20 Under 35 U.S.C. §103(a)

Applicant respectfully traverses the Examiner’s rejection of claims 6-7 and 17-20 under 35 U.S.C. 103(a) as being unpatentable over Kaminsky and Katz in view of Lazarus (previously cited).

As discussed hereinbefore, Kaminsky is not an effective reference against Applicant’s claims.

The combination of Lazarus and Katz (alone or in combination with any other cited references), does not disclose or suggest that which applicant claims with particularly. As previously set forth in Applicant’s Response dated Februarys 23, 2005, Lazarus is directed towards a system for selecting advertisements in a computer environment. The system includes a database of electronic advertisements. Observed behavior of a user computer in the computer environment is converted to a “behavior vector.” The behavior vector is compared to a group of “entity vectors” indicative of the ads, and entity vectors closely associated with the observed

behavior are identified. A selector accesses the database with the identified entity vector to select electronic ads to communicate to the user computer. Applicant respectfully submits that Lazarus does not teach or suggest, “a processor ... receiving said first data set from the one or more subscriber servers and said second data set from the one as more demographic databases, said processor including a rule processor receiving said first data set and said second data set and applying said first and second data sets to one or more rules to determine a score predicting behavior relating to said collected information identifying said user,” as particularly claimed by Applicant.

Again, there is no mention or suggestion in Lazarus of determining a score via a rule processor applying rules to a first and second data set. Specifically, Applicant applies rules via a rule processor to two distinct data sets, i.e., “a first data set of user information” and a “second data set of said third party information relating to targeted market segments.” Lazarus in fact teaches away from applying rules which “requires extensive knowledge about the targeted operating domain” (lines 21-22 of col. 3). Indeed, Lazarus states unequivocally that its system for selecting ads “does not require any rules for operation [but is instead based on] neural network techniques” (lines 53-55 of col. 5, emphasis added).

Applicant respectfully submits that claims 6-7 and 17-20 depend ultimately from one of claims 1 and 14 and are therefore distinguishable from Lazarus for at least the reasons set forth hereinbefore with respect to claims 1 and 14. Furthermore, the Examiner does not cite Lazarus to make up for the above-described deficiencies of Kaminsky and Katz, but instead cites Lazarus only in regard to features in dependent claims 6-7 and 17-20. Accordingly, Applicant submits that claims 6-7 and 17-20 are distinguishable from the applied combination of Kaminsky, Katz and Lazarus for reasons set forth hereinbefore with respect to claims 1 and 14, and that the rejection of claims 6-7 and 17-20 is therefore improper and should be withdrawn.

Applicant respectfully traverses the Examiner’s rejection of claims 15-16 under 35 U.S.C. 103(a) as being unpatentable over Kaminsky and Katz in view of Gerace (previously cited by the Examiner).

Gerace is directed toward software for targeting end users based on so-called psychographic profiles formed by recording the users’ computer activity. Categories of interest

and display format in each category are revealed by the profile, based on user viewing of so-called "agate information." Using the profile, advertisements are displayed to appropriately selected users. Gerace, like Katz, does not disclose or suggest applying rules to a first and second data set to determine a score.

From the foregoing, it is clear that Kaminsky is not prior art and that none of Lazarus, Katz nor Gerace alone or in any combination disclose or suggest each and every element of Applicant's claims. Consequently, the Examiner has not made a prima facie case of obviousness. Consequently, the rejection of the claims under 35 U.S.C. 103(a) is improper and should be withdrawn.

Finally, Applicant respectfully traverses the Examiner's rejection of claims 9-12 and 21-22 under 35 U.S.C. 103(a) as being unpatentable over Kaminsky and Katz in view of "Official Notice."

From the foregoing, it is clear that Kaminsky is not prior art and that Katz fails to disclose or suggest each and every element of claims 15-16 or any other claim(s). Consequently, the Examiner has not made a prima facie case of obviousness. Consequently, the rejection of the claims under 35 U.S.C. 103(a) is improper and should be withdrawn.

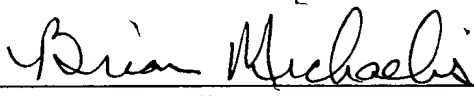
CONCLUSION

In view of the above, reconsideration and allowance of this application are now believed to be in order, and such action is hereby solicited. If any points remain in issue which the Examiner feels may be best resolved through a telephone interview, the Examiner is kindly requested to contact the undersigned at the telephone number listed below. The Examiner is invited and encouraged to telephone the undersigned with any concerns in furtherance of the prosecution of the present application.

Please charge any fee(s) that may be associated with this Response to Deposit Account No. 50-0369.

Respectfully submitted,

July 20, 2005
Dated:



Brian L. Michaelis, Reg. No. 34,221
Customer No. 21710
Attorney for Applicants
BROWN RUDNICK BERLACK ISRAELS LLP
One Financial Center, Box IP
Boston, MA 02111
Tel: 1-617-856-8369
Fax: 1-617 856-8201

1c973 U.S. PTO
01/09/01

01-10 81

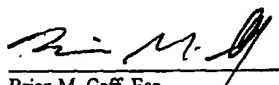
A prwl

Express Mail Mailing Label No. EL661684968US

PROVISIONAL APPLICATION FOR PATENT COVER SHEET	Attorney Docket No.:	PLP-001PR
	First Named Inventor:	David L. Kaminsky

1c9714 U.S. PTO
60/260583
01/09/01

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c)

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
David Louis	Kaminsky	103 Corbin Hill Circle, Chapel Hill, NC 27514
David Phillip	Johnson	310 Parkmeadow Drive, Apex, NC 27502
David Mark	Ogle	111 Jodhpur Drive, Cary, NC 27513
Thomas Owings	Rowe	106 Adams Way, Chapel Hill, NC 27516
Gregory Alexander Reasoner	DeKoenigsberg	618 Edinborough Drive, Durham, NC 27703
TITLE OF THE INVENTION (280 characters max)		
METHODS AND APPARATUS FOR EXCHANGING INFORMATION BETWEEN SERVERS AND CLIENTS		
ENCLOSED APPLICATION PARTS (check all that apply)		
<input checked="" type="checkbox"/> Specification (Number of Pages 28)	<input type="checkbox"/> CD(s) (Number)	
<input checked="" type="checkbox"/> Drawings (Number of Sheets 3)	<input checked="" type="checkbox"/> Application Data Sheet	
<input checked="" type="checkbox"/> Return Receipt Postcard	<input type="checkbox"/> Other (specify):	
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT		
<input checked="" type="checkbox"/> Applicant claims small entity status.		
<input checked="" type="checkbox"/> A check or money order is enclosed to cover the filing fees.		
<input type="checkbox"/> The Commissioner is hereby authorized to charge the required fees to Deposit Account No. 20-0531. Enclosed is a copy of this sheet.		
<input checked="" type="checkbox"/> The Commissioner is hereby authorized to charge any additional filing fees or credit any overpayment to Deposit Account Number 20-0531.		
FILING FEE AMOUNT \$		75.00
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.		
<input checked="" type="checkbox"/> No.		
<input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are: _____		
CORRESPONDENCE ADDRESS		SIGNATURE BLOCK
Direct all correspondence to: Patent Administrator Testa, Hurwitz & Thibault, LLP High Street Tower 125 High Street Boston, MA 02110 Tel. No.: (617) 248-7000 Fax No.: (617) 248-7100		Respectfully submitted,  Brian M. Gaff, Esq. Attorney for Applicants Testa, Hurwitz & Thibault, LLP High Street Tower 125 High Street Boston, MA 02110

This
provisional
60/260,583
filed
1/9/01

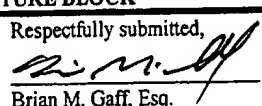
DOES
include
cited
portions
of
KAMINSKY!

VER 12/00

2004438-1

FEE TRANSMITTAL FY 2001

Complete if Known	
Application Serial Number:	To Be Assigned
Filing Date:	January 9, 2001
First Named Inventor:	KAMINSKY
Group Art Unit:	To Be Assigned
Examiner Name:	To Be Assigned
Attorney Docket No.:	PLP-001PR

METHOD OF PAYMENT	FEE CALCULATION (continued)																																																																																					
1. <input checked="" type="checkbox"/> Payment Enclosed: <input checked="" type="checkbox"/> Check <input type="checkbox"/> Money Order <input type="checkbox"/> Other	3. ADDITIONAL FEES <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Large Entity Fee (\$)</th><th>Small Entity Fee (\$)</th><th>Fee Description</th><th>Fee Paid</th></tr> </thead> <tbody> <tr><td>130</td><td>65</td><td>Surcharge - late filing fee or oath</td><td></td></tr> <tr><td>50</td><td>25</td><td>Surcharge - late provisional filing fee or cover sheet</td><td></td></tr> <tr><td>130</td><td>130</td><td>Non-English specification</td><td></td></tr> <tr><td>2,520</td><td>2,520</td><td>For filing a request for reexamination</td><td></td></tr> <tr><td>110</td><td>55</td><td>Extension for reply within first month</td><td></td></tr> <tr><td>390</td><td>195</td><td>Extension for reply within second month</td><td></td></tr> <tr><td>890</td><td>445</td><td>Extension for reply within third month</td><td></td></tr> <tr><td>1,390</td><td>695</td><td>Extension for reply within fourth month</td><td></td></tr> <tr><td>1,890</td><td>945</td><td>Extension for reply within fifth month</td><td></td></tr> <tr><td>310</td><td>155</td><td>Notice of Appeal</td><td></td></tr> <tr><td>310</td><td>155</td><td>Filing a brief in support of an appeal</td><td></td></tr> <tr><td>270</td><td>135</td><td>Request for oral hearing</td><td></td></tr> <tr><td>130</td><td>130</td><td>Petitions to the Commissioner</td><td></td></tr> <tr><td>50</td><td>50</td><td>Petitions related to provisional applications</td><td></td></tr> <tr><td>180</td><td>180</td><td>Submission of Information Disclosure Statement</td><td></td></tr> <tr><td>710</td><td>355</td><td>Filing a submission after final rejection (37 CFR 1.129(a))</td><td></td></tr> <tr><td>710</td><td>355</td><td>For each additional invention to be examined (37 CFR 1.129(b))</td><td></td></tr> <tr><td colspan="3">Other fee (Specify) _____</td><td></td></tr> <tr><td colspan="3">Other fee (Specify) _____</td><td></td></tr> </tbody> </table>	Large Entity Fee (\$)	Small Entity Fee (\$)	Fee Description	Fee Paid	130	65	Surcharge - late filing fee or oath		50	25	Surcharge - late provisional filing fee or cover sheet		130	130	Non-English specification		2,520	2,520	For filing a request for reexamination		110	55	Extension for reply within first month		390	195	Extension for reply within second month		890	445	Extension for reply within third month		1,390	695	Extension for reply within fourth month		1,890	945	Extension for reply within fifth month		310	155	Notice of Appeal		310	155	Filing a brief in support of an appeal		270	135	Request for oral hearing		130	130	Petitions to the Commissioner		50	50	Petitions related to provisional applications		180	180	Submission of Information Disclosure Statement		710	355	Filing a submission after final rejection (37 CFR 1.129(a))		710	355	For each additional invention to be examined (37 CFR 1.129(b))		Other fee (Specify) _____				Other fee (Specify) _____								
Large Entity Fee (\$)	Small Entity Fee (\$)	Fee Description	Fee Paid																																																																																			
130	65	Surcharge - late filing fee or oath																																																																																				
50	25	Surcharge - late provisional filing fee or cover sheet																																																																																				
130	130	Non-English specification																																																																																				
2,520	2,520	For filing a request for reexamination																																																																																				
110	55	Extension for reply within first month																																																																																				
390	195	Extension for reply within second month																																																																																				
890	445	Extension for reply within third month																																																																																				
1,390	695	Extension for reply within fourth month																																																																																				
1,890	945	Extension for reply within fifth month																																																																																				
310	155	Notice of Appeal																																																																																				
310	155	Filing a brief in support of an appeal																																																																																				
270	135	Request for oral hearing																																																																																				
130	130	Petitions to the Commissioner																																																																																				
50	50	Petitions related to provisional applications																																																																																				
180	180	Submission of Information Disclosure Statement																																																																																				
710	355	Filing a submission after final rejection (37 CFR 1.129(a))																																																																																				
710	355	For each additional invention to be examined (37 CFR 1.129(b))																																																																																				
Other fee (Specify) _____																																																																																						
Other fee (Specify) _____																																																																																						
2. <input checked="" type="checkbox"/> The Commissioner is hereby authorized to credit or charge any fee indicated below for this submission to Deposit Account No. 20-0531. <input type="checkbox"/> Required Fees (copy of this sheet enclosed). <input checked="" type="checkbox"/> Additional fee required under 37 CFR 1.16 and 1.17. <input checked="" type="checkbox"/> Overpayment Credit.																																																																																						
3. <input checked="" type="checkbox"/> Applicant claims small entity status.																																																																																						
FEE CALCULATION 1. FILING FEE <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Large Entity Fee (\$)</th><th>Fee Description</th><th>Fee Paid</th></tr> </thead> <tbody> <tr><td>710</td><td>Utility filing fee</td><td></td></tr> <tr><td>320</td><td>Design filing fee</td><td></td></tr> <tr><td>150</td><td>Provisional filing fee</td><td>150.00</td></tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Number Filed</th><th>Number Extra</th><th>Rate</th><th>Amount</th></tr> </thead> <tbody> <tr> <td>Total Claims - 20 =</td><td></td><td>x \$ 18.00 =</td><td></td></tr> <tr> <td>Independent Claims - 3 =</td><td></td><td>x \$ 80.00 =</td><td></td></tr> <tr> <td colspan="2">Multiple Dependent Claim(s), if any</td><td>\$270.00 =</td><td></td></tr> <tr> <td colspan="3" style="text-align: right;">TOTAL:</td><td>150.00</td></tr> <tr> <td colspan="3" style="text-align: right;">SMALL ENTITY DISCOUNT:</td><td>75.00</td></tr> <tr> <td colspan="3" style="text-align: right;">SUBTOTAL (1)</td><td>(\$ 75.00)</td></tr> </tbody> </table> 2. AMENDMENT CLAIM FEES <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Claims Remaining After Amend.</th><th>Highest No. Previously Paid For</th><th>Present Extra</th><th>Rate</th><th>Fee Paid</th></tr> </thead> <tbody> <tr> <td>Total</td><td>-</td><td>=</td><td>x \$ 18.00 =</td><td></td></tr> <tr> <td>Indep.</td><td>-</td><td>=</td><td>x \$ 80.00 =</td><td></td></tr> <tr> <td colspan="2"><input type="checkbox"/> First Presentation of Multiple Dep. Claim</td><td></td><td>+ \$270.00 =</td><td></td></tr> <tr> <td colspan="3" style="text-align: right;">TOTAL:</td><td>(\$)</td><td></td></tr> <tr> <td colspan="3" style="text-align: right;">SMALL ENTITY DISCOUNT:</td><td>(\$)</td><td></td></tr> <tr> <td colspan="3" style="text-align: right;">SUBTOTAL (2)</td><td>(\$0.00)</td><td></td></tr> </tbody> </table>	Large Entity Fee (\$)	Fee Description	Fee Paid	710	Utility filing fee		320	Design filing fee		150	Provisional filing fee	150.00	Number Filed	Number Extra	Rate	Amount	Total Claims - 20 =		x \$ 18.00 =		Independent Claims - 3 =		x \$ 80.00 =		Multiple Dependent Claim(s), if any		\$270.00 =		TOTAL:			150.00	SMALL ENTITY DISCOUNT:			75.00	SUBTOTAL (1)			(\$ 75.00)	Claims Remaining After Amend.	Highest No. Previously Paid For	Present Extra	Rate	Fee Paid	Total	-	=	x \$ 18.00 =		Indep.	-	=	x \$ 80.00 =		<input type="checkbox"/> First Presentation of Multiple Dep. Claim			+ \$270.00 =		TOTAL:			(\$)		SMALL ENTITY DISCOUNT:			(\$)		SUBTOTAL (2)			(\$0.00)		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: right;">SUBTOTAL (3)</td><td>(\$ 0.00)</td></tr> <tr> <td style="text-align: right;">SUBTOTAL (1)</td><td>75.00</td></tr> <tr> <td style="text-align: right;">SUBTOTAL (2)</td><td>0.00</td></tr> <tr> <td style="text-align: right;">SUBTOTAL (3)</td><td>0.00</td></tr> <tr> <td style="text-align: right;">TOTAL</td><td>(\$ 75.00)</td></tr> </table>	SUBTOTAL (3)	(\$ 0.00)	SUBTOTAL (1)	75.00	SUBTOTAL (2)	0.00	SUBTOTAL (3)	0.00	TOTAL	(\$ 75.00)
Large Entity Fee (\$)	Fee Description	Fee Paid																																																																																				
710	Utility filing fee																																																																																					
320	Design filing fee																																																																																					
150	Provisional filing fee	150.00																																																																																				
Number Filed	Number Extra	Rate	Amount																																																																																			
Total Claims - 20 =		x \$ 18.00 =																																																																																				
Independent Claims - 3 =		x \$ 80.00 =																																																																																				
Multiple Dependent Claim(s), if any		\$270.00 =																																																																																				
TOTAL:			150.00																																																																																			
SMALL ENTITY DISCOUNT:			75.00																																																																																			
SUBTOTAL (1)			(\$ 75.00)																																																																																			
Claims Remaining After Amend.	Highest No. Previously Paid For	Present Extra	Rate	Fee Paid																																																																																		
Total	-	=	x \$ 18.00 =																																																																																			
Indep.	-	=	x \$ 80.00 =																																																																																			
<input type="checkbox"/> First Presentation of Multiple Dep. Claim			+ \$270.00 =																																																																																			
TOTAL:			(\$)																																																																																			
SMALL ENTITY DISCOUNT:			(\$)																																																																																			
SUBTOTAL (2)			(\$0.00)																																																																																			
SUBTOTAL (3)	(\$ 0.00)																																																																																					
SUBTOTAL (1)	75.00																																																																																					
SUBTOTAL (2)	0.00																																																																																					
SUBTOTAL (3)	0.00																																																																																					
TOTAL	(\$ 75.00)																																																																																					
CORRESPONDENCE ADDRESS Direct all correspondence to: Patent Administrator Testa, Hurwitz & Thibault, LLP High Street Tower-125 High Street Boston, MA 02110 Tel. No.: (617) 248-7000 Fax No.: (617) 248-7100	SIGNATURE BLOCK Respectfully submitted,  Brian M. Gaff, Esq. Attorney for Applicants Testa, Hurwitz & Thibault, LLP High Street Tower-125 High Street Boston, MA 02110 Date: January 9, 2001 Reg. No.: 44,691 Tel. No.: (617) 310-8085 Fax No.: (617) 790-0332 Email: gaff@tth.com																																																																																					

Field of the Invention

Cross-Reference To Related Applications

- Serial Number 60/156,020; filed 23-Sep-1999.
- Serial Number 09/440,318; filed 12-Nov-1999.
- Serial Number 60/180,466; filed 4-Feb-2000.
- Serial Number 09/567,936; filed 10-May-2000.

[0003] There has been a rapid growth in networked computer systems, particularly those providing an end user with an interactive user interface. An example of an interactive computer network is the World Wide Web (hereafter, the “web”). The web is a facility that overlays the Internet and allows end users to browse web pages using a software application known as a web browser or, simply, a “browser.” Example browsers include Internet Explorer by Microsoft Corporation of Redmond, WA, and Netscape Navigator by Netscape Communications Corporation of Mountain View, CA. For ease of use, a browser includes a graphical user interface that it employs to display the content of “web pages.” Web pages are formatted, tree-structured repositories of information. Their content can range from simple text materials to elaborate multimedia presentations.

[0004] The web is generally a client-server based computer network. The network includes a number of computers (i.e., "servers") connected to the Internet. The web pages that an end user will access typically reside on these servers. An end user operating a web browser is a "client" that, via the Internet, transmits a request to a server to access information available on a specific web page identified by a specific address. This specific address is known as the Uniform Resource Locator ("URL"). In response to the end user's request, the server housing the specific web page will transmit (i.e., "download") a copy of that web page to the end user's web browser for display.

[0005] To ensure proper routing of messages between the server and the intended client, the messages are first broken up into data packets. Each data packet receives a destination address according to a protocol. The data packets are reassembled upon receipt by the target computer. A commonly accepted set of protocols for this purpose are the Internet Protocol (hereafter, "IP") and Transmission Control Protocol (hereafter, "TCP"). IP dictates routing information. TCP dictates how messages are actually separated in to IP packets for transmission for their subsequent collection and reassembly. TCP/IP connections are typically employed to move data across the Internet, regardless of the medium actually used in transmitting the signals.

[0006] Any Internet "node" can access a specific web page by invoking the proper communication protocol and specifying the URL. (A "node" is a computer with an IP address, such as a server permanently and continuously connected to the Internet, or a client that has established a connection to a server and received a temporary IP address.) Typically, the URL has the format `http://<host>/<path>`, where "http" refers to the HyperText Transfer Protocol, "host" is the server's Internet identifier, and the "path" specifies the location of a file (e.g., the specific web page) within the server.

[0007] As computing technology has evolved, users have been able to access increased amounts of information from an ever-expanding universe of data sources. Information from a myriad of sources is available to virtually anyone with a device that

is connected to a network and capable of "browsing" the latter. Despite advances in hardware and software, the sheer volume of information available can overwhelm an end user. An end user's attempts to access information relevant to his needs may fail because much of the material located by browsing may be generic or otherwise have little or no value to the end user.

[0008] Compounding this problem is the difficulty end users can have when attempting to recall the URLs of web pages of interest. Typically, an end user specifies a URL by typing its characters on the keyboard of the client device, thereby eliciting a response from a browser. Nevertheless, many URLs have become longer (i.e., use more characters) and counterintuitive. These attributes combine to increase the difficulty an end user can experience when attempting to access information of interest.

[0009] From the foregoing, it is apparent that there is still a need for a way that allows an end user to quickly and efficiently cull information of interest from the vast amounts of data available on a computer network, such as the Internet. Such a method should also promote the delivery of information that is tailored to the end user's needs or environment, or both.

Summary of the Invention

[0010] The present invention provides a way for end users to locate relevant information from the volume of data available on a computer network, such as the Internet. The information can be tailored to one or more of the end user's interests, needs, or environment, thereby reducing the potential of delivering generic, and potentially irrelevant, data. The invention simplifies the end user's interaction with the network by eliminating the need for manual entry of long or complicated URLs.

[0011] One embodiment of the invention features a method of providing information from a server to a client. As a first step, the client accepts a request for information from the end user. The client then constructs an identifier, such as a URL, that specifies the

source of the information. The information can be resident on the server (i.e., making the server the "source" of the information). Typically, though, the information resides on a web site or another network node. The identifier also includes a differentiating indicator that incorporates, for example, data identifying one or more of a device type, an end user, a user group, and a location. The client then transmits the identifier with the included (e.g. embedded) differentiating indicator to the server.

[0012] After receipt of the identifier, the server locates the information requested by the end user. The server then tailors (e.g., personalizes) the information for the end user based on the differentiating indicator and then transmits this tailored information to the client. The tailoring is performed by, for example, computing a score according to a predefined algorithm that operates on the differentiating indicator. This computation occurs at either the client location or the server location, and the value of the score determines, at least in part, the nature of the tailoring, for example, which information is selected for transmission to the client. This selection occurs at either the client location or the server location.

[0013] In a related embodiment, the invention provides an article of manufacture that includes a program storage medium having computer readable program code for causing a server to provide information to a client. The computer readable program code causes a computer to accept the request for information, construct the identifier with the included (e.g., embedded) differentiating indicator and transmit it to the server, tailor the information based on the differentiating indicator, and transmit the information from the server to the client. In a different embodiment, a program storage medium tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned delivery of tailored information to an end user.

[0014] In another embodiment, the initial request for information includes the step of interpreting a code associated with an article of commerce. The code is, for instance, a

code, such as a digital watermark, placed on the article as a form of identification. By way of example, the interpretation includes scanning the code using optical, radio frequency, magnetic or other methods. This code is then included in the identifier. To increase the accuracy of the interpretation, a data stream generated by the interpretation action (e.g., scanning) is examined to determine its rate and the presence of at least one preamble character that is typically included with the code. By assessing the data stream rate, or the presence of at least one preamble character, or both, the accuracy of the identifier may be improved, thereby improving the accuracy of the initial request for information.

[0015] In a further embodiment, the code associated with an article of commerce is compared with one or more known code types (e.g., code formats). This is accomplished by, for example, comparing the number of characters in the code with the number of characters in the known code type. Alternatively, the code checksum is compared with the checksum of the known code type. Irrespective of the method used, the differentiating indicator is then defined to be the code type that matches the code associated with an article of commerce.

[0016] In another embodiment, the server, after receipt of the identifier, maps the identifier to a "target URL". Because the identifier can be unique, the result of the server mapping the identifier can be a unique target URL. This URL is thus tailored to the particular end user. In other words, the server is still providing tailored information to the end user by providing the client with a "tailored pointer" (the unique target URL) to information. This method is as if the server itself was tailoring the information, and both methods are within the scope of the present invention. The server then transmits this target URL to the client. The client, typically using a browser, then accesses the target URL and displays the associated target file (e.g., the corresponding web page) to the end user. When the associated target file does not exist, the server accesses an error handling system. This error handling system intercepts an error generated by the server due to the missing target file. The system, for example, provides a new target URL to the client

that corresponds to a default content address. As an alternative, the system executes a root error handler program, the output of which specifies a new target URL that is also provided to the client. In any event, an objective of the error handling system is to redirect the end user to a new web page when the originally requested web page cannot be located. This redirection is typically seamless and generally performed without additional end user effort.

[0017] In a further embodiment, the target URL corresponds to an online form and the target URL also includes end user identification data that has been obtained from the differentiating indicator. The online form is responsive to the end user identification data. This allows, for example, the form to use the identification data to complete (i.e., "populate") fields in the form corresponding to the data without end user effort. To illustrate, an online form that includes a field for the user name extracts that user name from the end user identification data included in the target URL. The end user then sees the form with the user name already inserted. The end user is thus freed from manually entering his name.

[0018] In another embodiment, the client collects behavioral data based on the end user's activity and includes this data in the differentiating indicator. These data incorporate, for example, the frequency at which a specific code associated with an article of commerce is used in an end user's requests for information. Various codes associated with an article of commerce can also be categorized. Consequently, the frequency at which a specific code category is used in an end user's requests for information can be included in the behavioral data.

[0019] In a further embodiment, information is provided from the server to the client by the client first accepting a request for information from an end user by interpreting a code associated with an article of commerce. The client then constructs an encrypted identifier that specifies a source of the information. The encrypted identifier includes the code associated with an article of commerce (e.g., the latter is embedded in the former).

In its construction of the encrypted identifier, the client can perform the encryption or can use already encrypted data (e.g., incorporate an already encrypted code). The client then transmits this encrypted identifier to the server. The encrypted identifier is then mapped to a target URL without decryption. The server then transmits the target URL to the client.

[0020] In a related embodiment, the invention provides an article of manufacture that includes a program storage medium having computer readable program code for causing a server to provide information to a client. The computer readable program code causes a computer to accept the request for information by interpreting the code associated with an article of commerce, construct and transmit to the server the encrypted identifier with the included (e.g., embedded) code associated with an article of commerce, map the encrypted identifier to the target URL, and transmit the latter to the client. In a different embodiment, a program storage medium tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned delivery of information from a server to a client using the encrypted identifier.

[0021] In another related embodiment, the computer readable program code causes a computer to accept the request for information, construct the identifier and transmit it to the server, locate the information based on the identifier and transmit the information to the client if the information is found, or access the error handling system if the information is not found. A different embodiment includes a program storage medium that tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned actions.

[0022] In yet another embodiment, requests for information from a server to a client are logged. The client first accepts a request for information from an end user by interpreting a code associated with an article of commerce. The client then constructs an identifier that specifies a source of the information requested and includes the code. The

client transmits this identifier to the server, which maps the identifier to a target URL. The server then transmits this target URL to the client.

[0023] In addition to transmitting the target URL, the server typically responds with a header that includes data regarding the exchange with the client. In this embodiment, at least one portion of this header response is logged (e.g., stored) for further analysis. By way of example, the target URL is caused to appear in the server header response, and the logged portion of the response includes the target URL.

[0024] In a related embodiment, the invention provides an article of manufacture that includes a program storage medium having computer readable program code for causing a computer to log requests for information from a server to a client. The computer readable program code causes a computer to accept the request for information by interpreting the code associated with an article of commerce, construct and transmit to the server the identifier with the included (e.g., embedded) code associated with an article of commerce, map the encrypted identifier to the target URL, log at least one portion of the server header response, and transmit the target URL to the client. In a different embodiment, a program storage medium tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned logging of requests for information from a server to a client.

[0025] In another embodiment, the invention offers a way to select the service level provided by a server to a client. The server can provide, for example, a service level that includes static content, such as simple web page displays. Alternatively, the server can provide dynamic content, such as web pages supporting interactive electronic commerce. In any event, the needs of the client and the nature of the information the end user requests influence the type of service level the server provides.

[0026] In this embodiment, a request for information is accepted and the client then selects a content type. An identifier, such as a URL, that specifies a source of the information is then constructed. Within this identifier is a designator that specifies the

content type. This designator, which the client can determine, can also be determined at least in part from a code associated with an article of commerce. The identifier with the included (e.g., embedded) designator is then transmitted to the server and, after receipt, the server executes program instructions based on that designator. These program instructions cause the server to provide the selected content type (e.g., static or dynamic). The requested information is then transmitted from the server to the client using the selected content type.

[0027] In a related embodiment, the invention provides an article of manufacture that includes a program storage medium having computer readable program code for causing a computer to select the service level. The computer readable program code causes a computer to accept the request for information, select the content type, construct the identifier and include the designator within it, transmit it to the server, and execute program instructions based on the designator to select the service level. In a different embodiment, a program storage medium tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned selection of a service level.

[0028] A further embodiment features a way to include supplementary data in a request for information from a server. Initially, an identifier, such as a URL, that specifies a source of the information is constructed. The supplementary data are then compressed using a persistent compression. The compression is "persistent" because it resists decompression (or "expanding") by mechanisms such as URL encoding. The supplementary data, once compressed, are included in the identifier, which is then transmitted to the server.

[0029] In a related embodiment, the invention provides an article of manufacture that includes a program storage medium having computer readable program code for causing a computer to include the supplementary data in the request for information. The computer readable program code causes a computer to construct the identifier, compress

the supplementary data using a persistent compression technique and include it in the identifier, and transmit the identifier to the server. In a different embodiment, a program storage medium tangibly embodies a program of instructions executable by the computer to perform the corresponding method steps for the aforementioned inclusion of supplementary data in the request for information.

[0030] Other aspects and advantages of the present invention will become apparent from the following detailed description, taken in conjunction with the accompanying drawings, illustrating the principles of the invention by way of example only.

Brief Description of the Drawings

[0031] The foregoing and other objects, features, and advantages of the present invention, as well as the invention itself, will be more fully understood from the following description of various embodiments, when read together with the accompanying drawings, in which:

- Figure 1 is a schematic view of a hardware and software system constructed in accordance with an embodiment of the invention;
- Figure 2 depicts pseudo-code for an embodiment of the invention; and
- Figure 3 is a workflow diagram showing an embodiment of the general operation of the invention.

Detailed Description

[0032] As shown in the drawings for the purposes of illustration, the invention may be embodied in a method of quickly and efficiently delivering information to an end user tailored to the end user's specific needs, interests, and environment. A system according to the invention enriches the end user's browsing experience by providing desired information while reducing the likelihood of supplying irrelevant information. The invention avoids the problem of requiring the end user to enter long or complicated URLs to locate information of interest.

[0033] Figure 1 shows a representative client-server implementation of the invention 100 that includes a server 102 and a client 124, which communicate over a medium such as the Internet 120. In one embodiment, the components of server 102 intercommunicate over a main bi-directional system bus 104. The main sequence of instructions effectuating the invention reside on a mass storage device (such as a hard disk or optical storage unit) 106 as well as in a main system memory 108 during operation. Execution of these instructions and effectuation of some of the functions of the invention is accomplished by a central processing unit ("CPU") 110. Within the server 102, a network interface 118 is connected to the main bi-directional system bus 104. The server 102 is connected to the Internet 120 via the network interface 118 and a server connection 119.

[0034] The executable instructions that control the operation of the CPU 110 and thereby effectuate the functions of the invention are conceptually depicted as a series of interacting modules resident within the memory 108. (Not shown is the operating system that directs the execution of low-level, basic system functions such as memory allocation, file management and operation of the mass storage devices 106.) A web server software module 116 configures the server 102 as a web site, thereby conferring the capability of communicating over the web. Thus, web pages stored on the server 102 are made accessible across the web. As is well understood in the art, communication over the Internet 120 is accomplished by encoding information to be transferred into data packets. Each packet receives a destination address according to a consistent protocol, and each is reassembled upon receipt by the target computer. A commonly accepted set of protocols for this purpose includes the aforementioned Internet Protocol ("IP") and the Transmission Control Protocol ("TCP"). The Internet supports a large variety of information-transfer protocols, and the web represents one of these. Web-accessible information is identified by a URL, typically with the format described above.

[0035] Data exchange is typically effected over the web by means of web pages. In this case, the mass storage device 106 contains various aspects of the site web pages.

These aspects include, for example, formatting or mark-up instructions and associated data, and "applet" instructions that cause a properly equipped remote device, such as a computer, to present a dynamic display.

[0036] The request for information by the client 124 is typically accomplished using client software 132. The client software 132 constructs an identifier, in one embodiment in the form of a URL, and transmits the URL to the server 102 using the Internet 120. After receipt of this information, the server 102 first checks if it has a copy of that web page (i.e., a "cached copy") to be transmitted to the client 124. If a cached copy is not present, the server 102 transmits to the web site defined by the identifier (i.e., the URL), over the Internet 120, a request for a copy of the particular web page. Alternatively, if a cached copy is not present, the client can request for a copy of the particular web page. This is accomplished by the server 102 transmitting to the client 124 a pointer to the web site defined by the identifier (i.e., the URL). Using this pointer, the client then requests the particular web page from the web site via the Internet 120.

[0037] The process by which the server 102 locates the web page for transmission to the client 124 includes mapping the initial request for information to a target URL. A mapping engine 112, operating in conjunction with the web server software 116, performs this task. The mapping engine 112 receives the request for information and then constructs a new, target URL that defines the particular web page for retrieval and delivery to the client 124. Methods by which the mapping engine 112 constructs the target URL are disclosed in the aforementioned applications incorporated herein by reference. In brief, these methods include extracting data from the identifier to generate a file name and path of a file stored on the server 102. This file typically includes the target URL that the server 102 transmits to the client 124.

[0038] In one embodiment, the server 102 "personalizes" the web page to be transmitted to the client 124. This personalization includes accessing and interpreting data specific to the end user, such as demographics, that is typically embedded in the

identifier. The server 102 uses the data to, for example, augment the web page to be transmitted to the client 124. Alternatively, the server 102 uses the data to select a unique web page to be transmitted to the client 124. In one embodiment, the data are interpreted by mathematical manipulation to calculate a "score" that represents a profile of the end user. A server scoring engine 114, functioning with the web server software 116, performs this task. The mapping engine 112 then uses the value of the score to, for example, augment or select the web page to be transmitted to the client 124.

[0039] If the web page corresponding to the target URL cannot be located, an error handling system 115 (as described below) operates with the web server software 116 to provide the end user with alternative content. In one embodiment, the error handling system 115 selects alternative content that is related to the end user's request for information. This minimizes the potential of the end user receiving information that is not relevant to his initial request. To illustrate, assume the end user has requested information on a telephone manufacturer's particular model of wireless telephone. If the web page associated with the particular model cannot be located, the error handling system 115 then provides the end user with alternative content. This alternative content can include, for example, a web page that provides an overview of all wireless telephones produced by the telephone manufacturer. Thus, the end user receives information that is at least related to his initial request.

[0040] The data embedded in the initial request for information typically represent end user specific data, such as demographics. To increase the amount data sent to the server 102 without unduly burdening the network, the client 124 also includes a compression engine 136 that functions with (or may be part of) the client software 132. The compression engine 136 operates on the data before it is embedded in the initial request for information sent to the server 102. A purpose of the compression engine is to compress the data before including it in the initial request, thereby increasing the efficiency of the transmission. Furthermore, by providing additional end user specific

data beyond the minimum amount needed to tailor the particular web page, the server 102 can increase the degree to which it tailors the particular web page.

[0041] In a different configuration, the client 124 includes a client scoring engine 134 that functions with (or may be part of) the client software 132. Typically operating on end user specific data, the client scoring engine 134 modifies the initial request for information before the client 124 sends it to the server 102. These modifications help define the particular web page, performing a function akin to that performed by the server scoring engine 114, and eliminates the need for or increasing the effectiveness of the latter.

[0042] The client 124 receives from the server 102 via the Internet 120 the target URL or the alternative content provided by the error handling system 115. The client 124 typically includes a monitor 126 and a storage medium 130. A browser 128 running on the client 124 receives the transmission from the server 102 and displays a representation of the particular web page or the alternative content on the monitor 126.

[0043] A client connection 122 between the Internet 120 and the client 124, as well as the server connection 119 between the server 102 and the Internet 120, may take many forms. Typical examples include high-speed dedicated lines, a wireless link, as well as simple dial-up connections. With respect to a wireless link, a wireless client such as a personal digital assistant ("PDA") or wireless communication device is able to receive the benefit of the invention. It should also be noted that communication between the server 102 and the client 124 may be performed over a medium other than the Internet 120 as shown in Figure 1. Any network, such as an intranet or other proprietary communications medium may substitute for the Internet 120 and still remain within the scope of the present invention.

[0044] In brief overview, Figure 2 depicts pseudo-code associated with one embodiment of the invention. In this embodiment, the client accepts data from the end user and examines it to determine if it represents a (scanned) code associated with an

article of commerce. This determination is based on the rate at which the data is read and the presence of a preamble character in the data stream. If the rate is below a threshold value and the preamble character is not present, the client does not process the end user input as a code. Conversely, if the client determines the data does represent a code, the client then determines the appropriate service level. This is either "static" or "dynamic" and the client can make the determination based on the code itself.

[0045] In the next series of steps detailed in Figure 2, the client generates a differentiating indicator based on various parameters known by or provided to the client. Initially, the client compares the format of the code with the formats of known code types (e.g., UPC or ISBN codes). If the client determines the code matches a known code type, then the code type information is included in the differentiating indicator. Other parameters added to the differentiating indicator include a device type identifier (identifying the hardware in use), a user group identifier (identifying a group or class of end users), a location identifier (identifying a physical or geographic location), end user identification data, and end user behavioral data. Optionally, the client can score the end user identification data and the end user behavioral data as described herein. In this case, the client includes the score in the differentiating indicator.

[0046] After generating the differentiating indicator, the client then compresses it using the persistent compression technique described herein. The client then constructs an identifier by combining the code, content type, and differentiating indicator, and then transmits this identifier to the server. After receipt of the identifier, the server extracts the code, content type, and differentiating indicator. The server then decompresses the differentiating indicator and constructs a target file pointer. (Note that in alternative embodiments it is not necessary to decompress the differentiating indicator, because the compressed differentiating indicator may be used directly in the construction of the target file pointer.) This target file pointer includes the code and the content type. Optionally, if the server is performing a scoring operation in lieu of the client, then the server selects the proper segment of the differentiating indicator to score. This segment

includes, for example, the end user identification data, or the end user behavioral data, or both. The server then includes the score in target file pointer. (If the server is not performing scoring, then the differentiating indicator itself is included in the target file pointer.)

[0047] The server then maps the target file pointer to a target URL using, for example, methods described in the aforementioned applications incorporated herein by reference. If the target URL is not found, then an error handling system present on the server identifies alternative content (i.e., default content) that the server will provide to the end user. The server then redefines the target URL to associate it with the default content.

[0048] The server logs the information request and includes the target URL in the logged data. The server then transmits the target URL to the client. A browser program executing on the client receives the target URL and displays the associated web page for viewing by the end user.

[0049] Figure 3 shows a workflow diagram 200 that represents an embodiment of the invention. In greater detail, the client accepts a request for information from an end user (step 202). This information typically is to be delivered to the end user in the form of a web page. Optionally, the client can accept the request for information by detecting and interpreting a code associated with an article of commerce (step 204). This code, which can be placed on or in the article or its packaging, is detected and interpreted by using, for example, optical, radio frequency, magnetic, or other scanning techniques known in the art. Data related to this code, or the code itself, is then included in the request for information.

[0050] When the code is detected and interpreted (step 204), a data stream representing the code is typically generated by, for example, a scanning device used to read the code. In one embodiment, this data stream is accepted and examined to determine the presence of at least one preamble character. The preamble character is

typically used to signify the boundaries (i.e., the start, end, or both) of the code data. The data stream rate is also determined. This rate is related to the speed at which the code is scanned.

[0051] It is not uncommon for the data stream to use a transmission medium shared by other devices. For example, the scanning device and a computer keyboard may share a common data input port on the client. In this case, the client may have difficulty discriminating between data originating from the keyboard and data originating from the scanning device. Data from each of these sources typically require different processing, and the inability of the client to discern the origin of the data can lead to improper or unexpected operation of software executing on the client.

[0052] The presence of a preamble character or a data stream rate exceeding that typically achieved by manual entry (e.g., by typing on a keyboard) generally signifies the presence of code data, not data from another source, such as a keyboard. The interpretation step (step 204) looks for these attributes and thus classifies the data stream according to its origin. This ensures the data receives the proper processing, thereby increasing operational accuracy.

[0053] In step 212, in one embodiment the client constructs an identifier that specifies a source of the information requested by the end user and includes a differentiating indicator in the identifier. This identifier typically includes a URL. In another embodiment, the identifier can also include the code associated with an article of commerce. By combining the URL with end user specific data 208 resident on the client, the client creates the differentiating indicator in step 216, and then incorporates the differentiating indicator into the identifier. The differentiating indicator typically includes information specific to the end user, or hardware involved, or both. Examples include a device type identifier 218, which distinguishes the client hardware in use, and a location identifier 224, which designates its physical or geographic location. The differentiating indicator can also include end user identification data 226. Furthermore,

end users can be categorized or aggregated in to specific user groups, and the differentiating indicator can include a corresponding user group identifier 220.

[0054] To help distinguish between the various types of codes associated with an article of commerce, the client compares a received code with one or more known code types (e.g., formats) in step 206. The client receives the code using, for example, a scanner as described above or by manual entry by the end user's use of an input device, such as a keyboard. Typical known code types include the UPC and ISBN formats, which have standardized representations . When the comparison reveals a match to a known code type, that code type is defined to be the differentiating indicator. The client performs the comparison by comparing the number of characters in the received code with the number in the known code type. Alternatively, the client can perform the comparison by comparing the checksum of the received code with the checksum of the known code type.

[0055] The client can optionally collect behavioral data based on an end user's activity (step 230) and include it in the differentiating indicator (step 228). The behavioral data incorporates, for example, the frequency at which an end user's requests for information use a specific code. Various codes may also be categorized. Consequently, the frequency at which a specific code category is used in an end user's requests for information may be included in the behavioral data. For example, consider a case where there are three specific codes, "C1", "C2", and "C3". Assume the behavioral data collected by the client reveals the end user has made five requests for information using code C1, ten requests using code C2, and thirteen requests using code C3. In one embodiment, the system then constructs a URL of the form:

http://<host>/<code>/<path>/<filename>?&5&10&13

In this URL, <host> and <path> are as defined above, <filename> is the name of the file to be accessed, and <code> refers to the code associated with the request for information (e.g., a code associated with an article of commerce). The characters following the

question mark (i.e., a delimiter) are the frequencies of code use, positionally encoded within the URL.

[0056] After constructing the identifier (step 212), the client transmits it as a URL to a local server that retrieves and tailors the requested information (step 236). The local server can retrieve a copy of the requested information from a local cache (step 240), if such a copy exists. Alternatively, the local server can obtain the requested information from one or more other servers (step 238).

[0057] The local server tailors (e.g., personalizes) the requested information typically by examining the differentiating indicator. In one embodiment, the local server tailors the requested information by computing a score according to a predefined algorithm that operates on the differentiating indicator (step 242). The value of the score determines, at least in part, the information the local server selects to be returned to the end user in response to his request. The scoring algorithm can be, for example, a linear model where each item of data included in the differentiating indicator has an associated statistical "weight". In this model, each data item is multiplied by its associated weight and the resulting products are summed to calculate a score. To illustrate, consider the case where the differentiating indicator includes data (using, for example, the method of positional encoding described above) regarding the end user's demographics, such age and annual income. Assume the age of the end user is forty-five and his annual income is \$ 100,000. For this example, further assume the age range of thirty to fifty is represented by the code "3" and the income by the code "8". Lastly, assume the scoring algorithm is a linear model that gives a twenty percent weight to age and an eighty percent weight to annual income. Mathematically, this is represented by the equation $S = 0.2(3) + 0.8(8) = 7.0$, where "S" is the computed score. Thus, by examining the differentiating indicator and computing a score, the server uses the latter to categorize the end user based on his demographic data. Consequently, the server selects information to fulfill the end user's request that is also appropriate for that end user's demographics. The server accomplishes this by selecting information based on the value

of the score. Note that such scoring is not limited to operating on end user demographic data: the behavioral data described above may be similarly scored and the server may select information on this basis. In essence, scoring may be performed on virtually any amenable data included in the differentiating indicator, and the score used to select information in response to the end user's request.

[0058] In an alternative implementation, the client may perform substantially the same scoring function (step 232). In this case, the client includes the score within the identifier that it sends to the local server. Furthermore, the client, instead of the local server, may select the information to be returned to the end user (step 234). The client can base its selection on an examination of the score. The selection is reflected in the identifier constructed in step 212.

[0059] The degree to which the requested information is tailored is generally influenced by the amount of demographic data available. Increasing the amount of demographic data present in the differentiating indicator is one way to enhance the tailoring. Nevertheless, including too much information in the differentiating indicator can burden the communication links between the local server and the client. This can degrade network performance, leading to reduced local server responsiveness and a corresponding increase in end user frustration.

[0060] To minimize these problems without sacrificing the enhanced tailoring, an embodiment of the invention features a persistent compression technique. The technique is "persistent" because the compressed data (i.e., the output of the compression step 214) uses characters with byte sizes that do not expand when interacting with web servers. This expansion can occur, for example, if a web server uses a mechanism known as "URL encoding." This mechanism, typically adopted as a security measure, causes certain characters to be translated in to others having larger byte sizes. Consequently, the efficacy of the compression is compromised. The persistent compression technique ensures the characters representing the compressed data are those resistant to expansion.

These characters are typically alphanumeric and are used to represent, for example, numerical values that, in turn, represent the demographics. Because there are fewer alphanumeric characters than a range of values, the values are typically scaled by dividing by a base divisor. The quotient is associated with a non-expanding alphanumeric character. To illustrate, consider the case where values between 0 and 999 are to be encoded. Assume there are fifty available alphanumeric characters (e.g., the alphabet "A" through "Y", both upper and lower case) available that are unaffected by URL encoding. Dividing the number of values (1000) by fifty yields the base divisor, twenty. Each value to be encoded is divided by the base divisor and the integer portion of the quotient is translated in to the corresponding alphanumeric character. For example, to encode the value "991", the value is first divided by the base divisor (twenty) to yield an integer quotient of "49". Assume the letter "a" corresponds to "0", the letter "b" to "1", the letter "y" to 24, the letter "A" to 25, the letter "B" to 26, the letter "Y" to 50, etc. Thus, "991" is encoded as the letter "X". Furthermore, certain characters may be reserved to signify "large", "small" and "undefined" values. Thus, with knowledge of the base divisor, the local server can decompress the data and reconstitute the appropriate values. Note that because the integer portion is of the quotient is encoded, the local server may not be able to reconstitute the exact value. To illustrate using the example above, the local server would multiply the value corresponding to "X" (49) by the base divisor (20), yielding 980. Although this is not equal to the original data (991), the deviation may not be significant, particularly when the data are categorized into various ranges (e.g., 20 through 40, 250 through 275, etc.). This is not uncommon when the data represents demographics.

[0061] In this embodiment, supplementary data, such as demographics, are compressed using the persistent compression technique (step 214) and included in an identifier, such as a URL, that the client constructs (step 212). This identifier with the compressed supplementary data is then transmitted to the server.

[0062] In one embodiment of the invention, the local server receives the end user's request for information and maps it to a target URL (step 244), tailored as described above. The mapping step 244 typically uses methods disclosed in the aforementioned applications incorporated herein by reference. Briefly, these methods include extracting data from the identifier constructed in step 212 to generate the file name and path of a file stored on the local server. This file (the "pointer file") typically contains the target URL. This target URL is associated with a target file (e.g., the web page corresponding to the target URL). If this target file exists, it is transmitted from the local server to the client and displayed for the end user (step 254). If the pointer file, the target file, or both do not exist or cannot be found, the local server accesses an error handling system (step 248). The error handling system intercepts any server-generated error and defines the target URL to correspond to a default content address (step 252). In this way, default content (e.g., another web page known to exist) is provided to the end user in the place of a display of the error condition. Alternatively, the error handling system can execute a root error handler program (step 250). This program systematically searches for a number of different web sites to transmit to the client to replace that site represented by the target URL. The output of the root error handler program includes a URL that defines the replacement site. Irrespective of the method used, an objective of executing the error handling system (step 248) is to shield the end user from error conditions that would likely cause considerable frustration. Furthermore, the error handling system and the root error handler program may be configured to identify the default content and the replacement site, respectively, that are related to the end user's request. This minimizes the presentation of irrelevant material to the end user.

[0063] A variation of this embodiment features the client accepting the end user's request for information (step 202), constructing the identifier that specifies a source of the information (step 212), and transmitting the identifier to the local server that locates the information based on the identifier (step 236). (The differentiating indicator can optionally be included in the identifier (step 216).) The error handling system is accessed when the server fails to locate the information based on the identifier (step

248). This includes the use of the default content address (step 252) or the execution of a root error handler program (step 250), both described above. Conversely, when the server locates the information based on the identifier, it transmits that information to the server.

[0064] Optionally, the local server logs (i.e., stores) the requests for information (step 246). This is typically done in conjunction with the mapping step (step 244), when the target URL is determined. In brief, when the client transmits its request for information to the local server, the response of the latter typically includes a “header” that has data regarding the exchange with the client. (An example of this is the header generated by the Apache Web Server software.) The local server is configured to eliminate certain items from the header and replace them with other data, including the target URL determined in step 244. The local server is further configured to extract specific items (i.e., log a portion of these header response), such as the target URL, from this header and insert them into a log file that the local server maintains. This log file can be examined to assess, for example, the type, nature, and frequency of requests for information, thereby providing a “data warehousing” functionality.

[0065] In another embodiment, the invention features a way of selecting the service level provided by a server to a client. Service level is defined, at least in part, by a content type. A content type pertains to the target URL, or the operation of the local server, or both, and is typically “static” or “dynamic”. This reflects the nature of the web page associated with the target URL (i.e., a static or dynamic web page). In this embodiment, the client accepts a request for information from an end user (step 202), selects a content type (step 210), and constructs an identifier (step 212) that includes a designator that specifies the content type. The identifier typically includes a URL. The client can determine this designator, which can further be determined, at least in part, by a code associated with an article of commerce.

[0066] The identifier with the included designator is then transmitted to the local server, which, in turn, executes program instructions based on the designator. These program instructions cause the local server to provide the selected content type (e.g., static or dynamic). The program instructions trigger the delivery of a static web page, or initiate the execution of one or more programs to provide the dynamic content. The information requested by the end user is then transmitted from the local server to the client using the selected content type.

[0067] Under certain conditions, it may be desirable to encrypt some or all of the identifier constructed in step 212 before its transmission to the local server. The client can perform the encryption during step 212. Alternatively, already encrypted data, such as an already encrypted code associated with an article of commerce that is part of the request for information, may be included within the identifier for transmission, as is, to the local server. The local server then maps the encrypted identifier to a target URL (step 244), and this is typically done without decryption. The local server then transmits the target URL to the client to display the associated target file for the end user (step 254).

[0068] In some instances, the target URL may correspond to an online form that the end user would typically complete by entering, for example, personal identifying data. Typically, an online form is responsive to data contained in its URL. This means the online form extracts data included in its URL and inserts that data into the corresponding fields or locations in the online form (i.e., "populates" the form). For example, characters representing the end user's name may be included in the URL. The online form then extracts those characters and, after performing any required processing, inserts them into a "name" field or location in the online form. Consequently, data to complete the online form are transferred into the latter without the need for the end user to enter the information. This streamlines the end user's interaction with the online form.

[0069] When the differentiating indicator (which is part of the identifier) includes the end user identification data 226, the latter can be extracted from the identifier and inserted into the target URL. Thus, when the target URL corresponds to an online form, responsive as described above, the end user identification data 226 populate the corresponding fields or locations in the form. Note that the online form may call for a particular format for the end user identification data 226. In this case, the local server reformats the latter as required after extracting it from the identifier and before inserting it into the target URL.

[0070] Note that because Figure 1 and Figure 3 are block diagrams, the enumerated items are shown as individual elements. In actual implementations of the invention, however, they may be inseparable components of other electronic devices such as a digital computer. Thus, many of the actions described above may be implemented in software that may be embodied in an article of manufacture that includes a program storage medium.

[0071] Another embodiment of the invention features a system to provide information from a server 102 to a client 124 in response to a request from an end user. In this system, the client 124 includes a user request interface 138 that receives the end user's request for information. An identifier constructor 140 communicates with the user request interface 138. This identifier constructor 140 assembles an identifier, typically a URL, which specifies a source of the requested information. The identifier constructor 140 inserts a differentiating indicator in the identifier. This differentiating indicator, as described above, includes information specific to the end user, the hardware involved, or both. The client 124 also includes a transmitter 142 that is in communication with the identifier constructor 140. The transmitter 142 then sends the identifier (with the included differentiating indicator) from the client 124 to the server 102.

[0072] In this embodiment, the server 102 includes an information tailoring apparatus 111 that, in response to the differentiating indicator, tailors (i.e., personalizes) the information that will be provided to the end user. The server has a transmitter 113 that is in communication with the information tailoring apparatus 111 and the client 124. The transmitter 111 sends the information from the server 102 to the client 124 for display for the end user.

[0073] In a different embodiment, the client 124 includes a code interpreter 144 that accepts a request for information from an end user. It accomplishes this by interpreting a code associated with an article of commerce. This code is as described above. An identifier constructor 140 communicates with the code interpreter 144 and assembles an identifier. Optionally, the identifier can be encrypted. The identifier includes the code associated with an article of commerce. A transmitter 142 sends this identifier to the server.

[0074] After the server 102 receives the identifier, a mapping engine 112 maps it to a target URL. Mapping is accomplished using the techniques described herein. The server 102 then transmits the target URL to the client using a transmitter 113 that is in communication with the mapping engine 112 and the client 124.

[0075] The server 102 can also include a usage monitor 117 that is in communication with the mapping engine 112. As requests for information are made of the server 102, this usage monitor 117 logs at least a portion of the server header response to these requests. The server 102 uses a locator 109 to find the information that the end user requested. In some instances, the server 102 may be unable to locate this information. To address such situations, the server 102 includes an error handling system 115 that is in communication with the locator 109. The error handling system 115 executes an error handling routine that provides the end user with alternative content when the requested information is not available or cannot be found. The server 102 includes a transmitter 113 that is in communication with the locator 109 and the client

124. The transmitter 113 sends the client 124 the requested information (if the latter is located) and the alternative content (if the requested information cannot be located).

[0076] In a different embodiment, the client 102 includes a selector 146 that is in communication with the user request interface 138. The selector 146 chooses either a static or a dynamic content type. The identifier constructor 140 constructs an identifier that includes a designator that specifies the content type. When the server 124 receives the identifier, it executes program instructions depending on the designator (e.g., different instructions corresponding to a static content type versus different instructions corresponding to a dynamic content type).

[0077] The client 124 can also include supplementary data in its request for information from the server. A compression engine 136 that is in communication with the identifier constructor 140 compresses this supplementary data. The persistent compression technique described above is used.

[0078] From the foregoing, it will be appreciated that the methods provided by the invention afford a simple and effective way to locate information on a network, such as the Internet, that is of interest and tailored to an end user. The problem of an end user being unable to locate such information from large volume of data expeditiously is largely eliminated.

[0079] One skilled in the art will realize the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The foregoing embodiments are therefore to be considered in all respects illustrative rather than limiting of the invention described herein. Scope of the invention is thus indicated by the appended claims, rather than by the foregoing description, and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced therein.

Methods and Apparatus for Exchanging Information Between Servers and Clients

Abstract of the Disclosure

Methods and apparatus for providing information (e.g., web page content) to a client tailored to specific end users by including end user- or session-specific identity data in the initial query. The identity data are obtained from reading a code, such as a barcode, from a physical object, combining the code with information stored within the client, and embedding the result in the URL that forms the initial query. Before transmitting the query, the data can be compressed using a technique that resists inadvertent expansion, such as that caused by URL encoding. Once a server receives the query, it operates on the identity data using, for example, a scoring algorithm and maps the query to a target URL that specifies the location of the tailored information. If the file associated with the target URL cannot be located, the server accesses an error handling system to provide the end user with alternative content. The server logs the query, including the associated target URL, providing a data warehouse capability.

GAFFBM71574.2004106_1



Softwares

Download

Purchase

Examples

Support

About Us



Photomatix

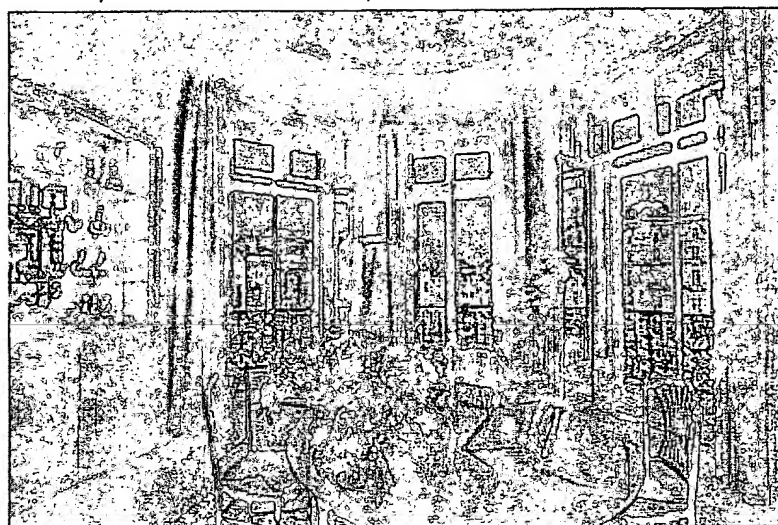
Increase the dynamic range of your photographs



1/125 s

1/15 s

1/2 s



Tone Mapping result on 32-bit image created from the 3 top photos

If you have ever photographed a high contrast scene, you know that selecting the correct exposure will not avoid blown out highlights and flat shadows. Photomatix Pro offers two ways to solve this problem:

- > **32-bit Tone Mapping:** Creates a High Dynamic Range Image from multiple exposures, then tones map it to compress its tonal range while preserving local contrast.
- > **Exposure Blending:** Merges differently exposed photographs into one image with increased dynamic range.

The result is an image that shows local details in both highlights and shadows.

[\[Learn more in the HDR Image FAQ\]](#)

- > [View examples](#) of images produced with Photomatix Pro
- > [Features overview + screenshots](#)
- > [Download a free trial version](#)
- > [Buy now!](#) [15-day money back guarantee](#)
- > See how [Exposure Blending](#) works

Latest news

14-July-05: Photomatix SDK

The Photomatix SDK provides a Dynamic Link Library and API documentation for functions such as High Dynamic Range image creation and merge of differently exposed images. Intended for software developers and for internal use only. [\[Read more\]](#)

6-July-05: Photomatix Pro v2.1 is released

Version 2.1 adds a faster and more robust automatic alignment tool, batch conversion of single files and High Dynamic Range Image histogram.

03-May-05: Comparison with Photoshop CS2

We did a few tests to [compare](#) the HDR Conversion of Photoshop CS2 with our Tone Mapping tool.

Photomatix Pro is a stand-alone program that runs on Mac OS X and Windows 98/Me/2000/XP. One license for Photomatix Pro costs US\$99.

The benefits of using Photomatix Pro include:

> Saving on lighting equipment

Given that most digital cameras can auto-bracket at different exposures, you do not need to acquire expensive lighting equipment -and carry it- when shooting high contrast scenes. Just enable Auto Exposure Bracketing, and let Photomatix merge the shots into an image with extended dynamic range.

28-Dec-04: Photomatix Basic released

Photomatix Basic is a freeware that creates High Dynamic Range (32-bit) images from multiple exposures and blends two differently exposed photographs.

15-Oct-04: PTConvertInterface 0.1

PTConvertInterface is a very simple utility that runs PTConvert, a command line program that converts HDR images in Radiance format (.hdr) into Floating Point JPEG images (.fjpg) that can be viewed as HDR panoramas in the latest version of PTVIEWER. [\[Read more\]](#)

[Subscribe](#) to our announcements

> Saving time in post-processing

Photomatix Pro is designed for productivity -- automatic blending, unlimited stacking, easy comparison of results and batch processing save hours of masking and layers work in image editing softwares.

> Taking advantage of your 32-bit images

Have you created a 32-bit HDR image in Photoshop CS2 and could not get a good HDR conversion? The tone mapping tool of Photomatix Pro may help. See how it [compares](#) to Photoshop CS2 HDR conversion and how it better preserves the local contrast of your image.

> Great pictures on cloudy days

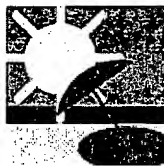
A shadowless hazy sunlight or an overcast sky usually result in dull-looking photographs. The tone mapping tool of Photomatix Pro can turn them into great-looking images. Check images [2](#) and [7](#) of our sample page for examples.

> Noise reduction

The Exposure Blending functions of Photomatix Pro merges together any number of bracketed photos -- this process is equivalent to image stacking which tends to reduce noise in the resulting image.

> Well exposed panoramas

A panoramic scene is almost always a high contrast scene -- you can't limit your view to only areas with the same brightness when shooting a 360° panorama. By taking views under several exposures and processing them in Photomatix Pro, your panorama will show details in both the dark and bright areas of the scene.



Photomatix

Details in Highlights & Shadows

Note: the name of our dynamic range increase software is *Photomatix* not *Photomatrix*.

[Softwares](#) | [Download](#) | [Purchase](#) | [Examples](#) | [Support](#) | [About Us](#)

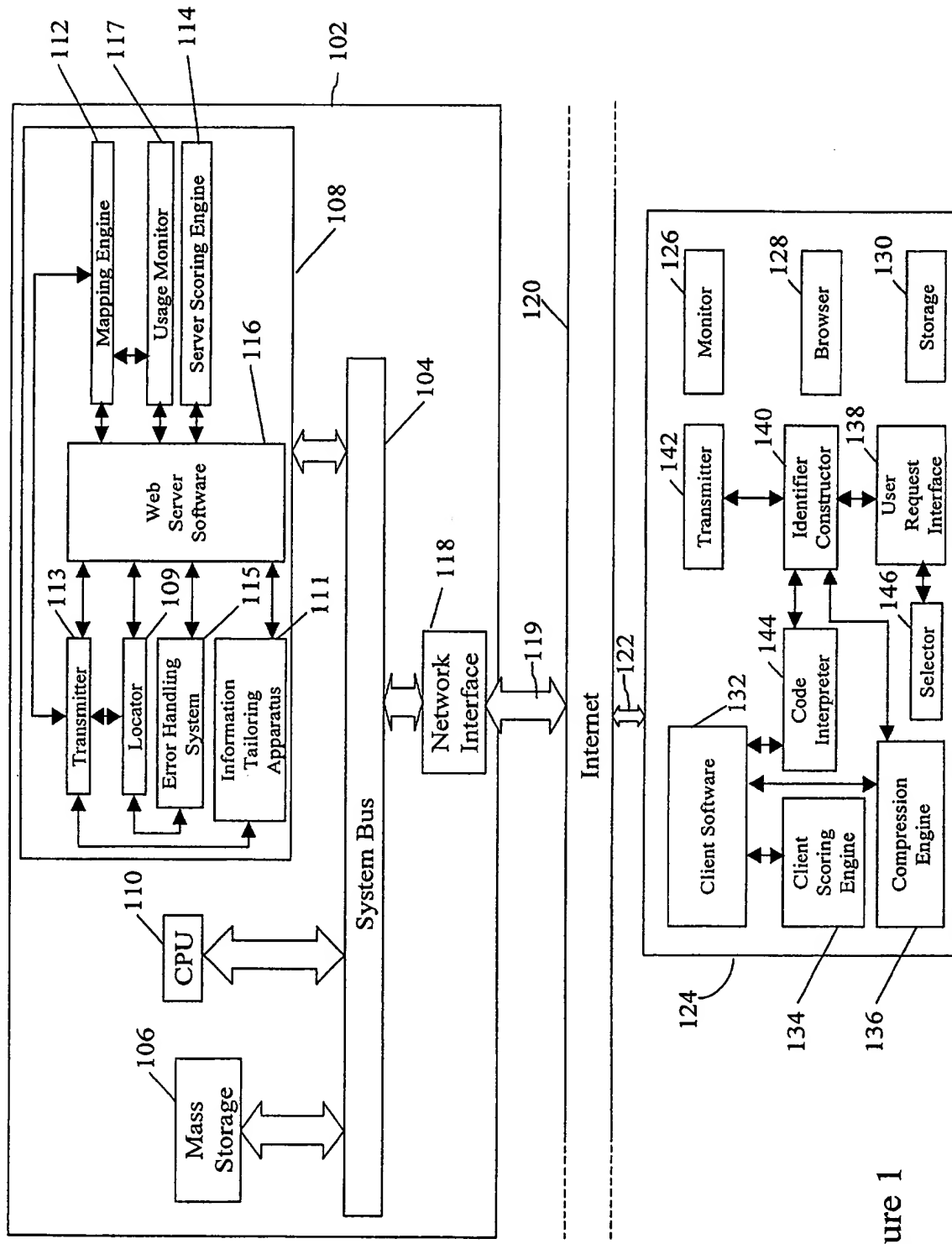


Figure 1

Figure 2

Figure 2

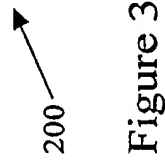


Figure 3

METHODS AND SYSTEMS FOR FACILITATING
USER ACCESS TO WEB SERVERS
USING PORTAL DEVICE

AN APPLICATION FOR
UNITED STATES LETTERS PATENT

By

Brent M. Kleinheksel
Durham, North Carolina

David L. Kaminsky
Chapel Hill, North Carolina

David M. Ogle
Cary, North Carolina

Gregory R. DeKoenigsberg
Zebulon, North Carolina

Kenneth G. Mages
Highland Park, Illinois

Chun-Wen Lai
Cary, North Carolina

This provisional
60/156,020
filed 9/23/99

Does ~~not~~
include

cited
portions

OF

KAMINSKY!

60156020-02095109

-1- "Express Mail" mailing number EJ35368816545
Date of Deposit 9/23/04
I hereby certify that this paper or fee is being deposited with
the United States Postal Service "Express Mail Post Office
to Addressee" service under 37 C.F.R. 1.10 on the date
indicated above and is addressed to the Commissioner of
Patents and Trademarks, Washington, D.C. 20231
Gregory A. Hunt, Registration No. 41,085
Gregory A. Hunt

Description

METHODS AND SYSTEMS FOR FACILITATING USER ACCESS TO WEB
SERVERS USING PORTAL DEVICE

Technical Field

5 The present invention relates to methods and systems for facilitating
user access to web servers. More particularly, the present invention relates to
methods and systems for facilitating user access to web servers using a portal
device.

Background Art

10 As the number of web servers connected to computer networks, such
as the Internet, increases, it becomes more and more difficult for users to
access their favorite web servers. In addition, the number of "hits" to each web
server decreases, because users are faced with a greater number of choices
as the number of web servers increases. As a result, company revenue
15 generated through web server hits decreases.

 One conventional mechanism for facilitating user access to web servers
and for increasing the number of hits to web servers is a software portal. A
software portal is a web server that includes links to other web servers. A
software portal can be accessed by users using a conventional web browser.
20 Once the user contacts the software portal, icons for various web servers are

552260-02095109

displayed in the browser window. The user accesses the web servers by clicking on one of the icons.

While software portals provide an interface through which users can access multiple web servers, the user must still manually invoke a browser and either manually or automatically, through a "favorites" interface of the browser, access the web server corresponding to the software portal. Each step required to be performed by users in order to access a desired web server increases the time required to access a web server and decreases the likelihood that the user will access the web server.

Some keyboards currently being sold for personal computers include special function keys that correspond to functions normally provided by browsers. For example, one conventional keyboard includes forward, back, and reload keys that allow users to access web servers according to the list of Hypertext Transfer Protocol (HTTP) queries displayed in the browser's address window. This feature is useful in allowing users to access previously-accessed web servers. However, the address window can include many queries corresponding to web servers that the user does not desire to access. As a result, the user can be required to "page" through multiple unwanted web servers before accessing a web server of interest. This paging increases the time required to access a web server of interest and decreases the likelihood that a user will access a web server of interest.

Accordingly, there exists a long felt need in the industry for facilitating user access to web servers and for increasing hits to selected web servers.

Disclosure of the Invention

The present invention includes methods and systems for facilitating user access to web servers using hardware portal devices and for increasing hits to selected web servers. As used herein, the term "web server" includes a server application executing on a computer connected to other computers over a network for providing data to client applications on the other computers in response to queries from the client applications. The data transmitted from the web server to the client applications can be used by the client applications to display a website to the user.

As used herein, the term "portal device" includes a device capable of connecting to a computer to facilitate user access to web servers and to facilitate collection of user information. The portal device can include an electrical or electro-mechanical user interface, such as a keypad, for receiving input from a user. Processing and communication circuitry identifies keys pressed by the user and communicates key codes for identifying the keys to downstream software. A mapping module generates a signal or code for varying the association between keys on the keypad and web servers.

A method for facilitating user access to web servers using a portal device includes transmitting a key code from a portal device to portal client software executing on a portal client computer. The portal client software forwards the key code along with user identification information to portal server software executing on a portal server computer. The portal server software determines a web server name that the user desires to access by accessing user information in a portal server database. The key code and the user

identification code can be used to perform a lookup in the database and extract the corresponding web server name. The portal server software forwards the web server name to the portal client software. The portal client software then formulates and sends a query to the web server specified by the web server name. The web server sends data back to the portal client device for display to the user. Thus, user information is collected and a website is accessed merely by pressing a single key.

Accordingly, it is an object of the present invention to provide novel methods and systems for facilitating user access to web servers.

It is another object of the present invention to increase hits to web servers.

It is another object of the present invention to provide a hardware portal for facilitating user access to web servers.

It is yet another object of the present invention to provide a method for collecting user preference information with regard to web server access.

It is yet another object of the present invention to provide business methods using hardware portal devices to increase hits to web servers.

These and other objects are achieved in whole or in part by the present invention. Some of the objects of the invention having been stated hereinabove, other objects will become evident as the description proceeds, when taken in connection with the accompanying drawings as best described hereinbelow.

Brief Description of the Drawings

The present invention will now be explained with reference to the accompanying drawings of which:

Figure 1 a block diagram illustrating a system for facilitating user access
5 to web servers using a portal device according to an embodiment of the present invention;

Figure 2 is a block diagram illustrating a portal device according to an embodiment of the present invention;

Figure 2(a) is a block diagram illustrating a user interface for a portal
10 device according to an embodiment of the present invention;

Figure 2(b) is a block diagram illustrating a mapping module for a portal device according to an embodiment of the present invention;

Figure 3 is a block diagram illustrating portal client software according to an embodiment of the present invention;

Figure 4 is a block diagram illustrating portal server software according
15 to an embodiment of the present invention;

Figure 5 is a partial block/partial flow diagram illustrating a method for facilitating user access to web servers according to an embodiment of the present invention;

Figure 6 is a partial block/partial flow diagram illustrating a method for
20 facilitating user access to web servers according to another embodiment of the present invention;

50156020-092399

Figure 7 is a partial block/partial flow diagram illustrating a method for facilitating user access to web servers according to yet another embodiment of the present invention;

Figure 8 is a perspective view of a portal device according to an embodiment of the present invention; and

Figure 9 is a top view of exemplary cards associated with a mapping module of a portal device according to an embodiment of the present invention.

Detailed Description of the Invention

Figure 1 is a block diagram illustrating a system for facilitating user access to web servers utilizing a portal device according to an embodiment of the present invention. In Figure 1, the system includes a portal device 100, portal client software 102 that executes on a portal client computer 104, and portal server software 106 that executes on a portal server computer 108. The portal client computer 104 and the portal server computer 108 may each comprise general purpose computers, such as IBM, IBM-compatible, Macintosh, or Macintosh-compatible computers. The portal client computer 104 and the portal server computer 108 can be connected to each other and to web servers 110 through a computer network, such as a TCP/IP network or a UDP/IP network. The portal device 100, the portal client software 102, and the portal server software 106 cooperate to facilitate user access to the web servers 110. Each of these components will now be discussed in more detail.

60156020 "092349"

Portal Device

The portal device **100** can comprise any device that receives input from a user indicative of a web server or a plurality of web servers that the user desires to access and transmits signals to the portal client software **102** indicative of the user's selection. Figure 2 is a block diagram illustrating a portal device **100** according to an embodiment of the present invention. In Figure 2, the portal device **100** includes a user interface **200** adapted to receive user input relating to a web server or a plurality of web servers that the user desires to access. In a preferred embodiment of the invention, the user interface **200** comprises a keypad having a plurality of keys, where each key is associated with a web server or a plurality of web servers that the user may desire to access. In an alternative embodiment of the invention, the user interface **200** can comprise a touch screen having areas associated with web servers that the user may desire to access.

The portal device **100** includes processing and communication circuitry **202** for monitoring the user interface **200** to identify a key or area being pressed or touched by the user and output a signal indicative of the key to the portal client computer **104** illustrated in Figure 1. The processing and communication circuitry **202** can comprise hardware, software, or a combination of hardware and software configured to communicate with the user interface **200** and the portal client computer **104**. For example, the processing and communication circuitry **202** can comprise a microprocessor programmed to perform the processing and communication functions described herein. An exemplary microprocessor suitable for use as the processing and

communication circuitry 202 is the Model No. CY631101A available from Cypress Semiconductor.

In order to communicate with the portal client computer 104, the processing and communication circuitry 202 can include a serial interface, such as an RS-232 interface, for communicating with the serial port of the portal client computer 104, a parallel interface for communicating with the parallel port of the portal client computer 104, or high speed serial interface for communicating with the USB port of the portal client computer 104. Alternatively, the processing and communication circuitry 202 may be configured to communicate with the portal client computer 104 through a wireless interface, such as an infrared interface or a radio interface.

The mapping module 204 can comprise hardware, software, or a combination of hardware and software that produces output signals or codes for varying the association between at least some of the keys or areas associated with the user interface 200 and web servers. The association between other keys associated with the user interface 200 and web servers may not be variable using the mapping module 204.

In one embodiment, the mapping module 204 comprises a card reader adapted to read information stored on cards insertable in the card reader. For example, the mapping module 204 can comprise a bar code reader. A card (not shown) insertable in the bar code reader can include a bar code that stores a card identification code. The card identification code instructs software downstream from the card reader to interpret user input received through the user interface 200 in a certain way. For example, the card

identification code can be used to vary the association between keys or areas associated with the user interface and web servers. For example, if the "news" key or area is pressed and an ABC News card is inserted in the card reader, software downstream from the card reader can generate a query to a web server having the domain name abcnews.com. If, the "news" key is pressed a CNN card is inserted in the card reader, software downstream from the card reader can generate a query to the web server having the domain name cnn.com. If no card is inserted in the card reader, software downstream from the card reader preferably sends a query to a default location, as specified by the key or area being pressed and any user preferences, as will be discussed in more detail below.

Exemplary User Interface and Detector Circuitry

Figure 2(a) illustrates exemplary user interface and detector circuitry that can be included in the user interface 200 and the processing and communication circuit 202. In Figure 2(a), the user interface 200 comprises a keypad 220 including a plurality of keys 222. Each of the keys 222 includes an indicium associated with a web server a plurality of web servers. For example, the indicium on each key can include a company identifier, a website identifier, a product or service category identifier, or other indicium for facilitating user access to a web server or a plurality of web servers. The user interface 200 can include any electrical, mechanical, electromechanical, or other mechanism for detecting when a user presses a key and for determining which key has been pressed. In the illustrated embodiment, the user interface 200 includes a matrix of row conductors 224 and column conductors 226 that underlies the

keys on the keypad. A plurality of switches $S_1 - S_n$ can be connected between the row conductors 224 and the column conductors 226. For example, there can be a single switch corresponding to each key 222 on the keypad 220. The row conductors 224 are electrically isolated from the column conductors 226 until one of the keys 222 is pressed.

A detector 228, which can be included in processing and communication circuitry 202, can be connected to the row conductors 224 and the column conductors 226 for detecting closure of one of the switches $S_1 - S_n$, identifying the key being pressed. The detector 228 can be variously configured. For example, the detector 228 can comprise hardware, software, or any combination of hardware and software that scans the keypad matrix to identify a key being pressed. In order to scan the matrix, the detector 228 can be adapted to sequentially or randomly generate electrical signals on the row conductors or the column conductors. The detector 228 can also be adapted to receive return signals from the row conductors or the column conductors when a key is pressed. If the detector 228 is adapted to transmit signals on the row conductors 224, then the detector 228 is preferably adapted to receive signals on the column conductors, and vice versa. In the illustrated embodiment, the detector 228 drives a transmitter 230 connected to the row conductors 224 for transmitting scanning signals on the row conductors 224. The detector 228 monitors signals output from a receiver 232 connected to the column conductors 226 for receiving return signals from the column conductors 226.

When a key is pressed by the user, the switch corresponding to the key being pressed is closed. The transmitter 230 sequentially or randomly transmits signals on the row conductors 224 under control of the detector 228.

When the transmitter 230 transmits a signal on the row conductor connected to the switch corresponding to the key being pressed, the receiver 232 detects the return signal on the column conductor connected to the switch and forwards the return signal to the detector 228. Because the detector 228 knows the row conductor on which the transmission occurred and the column conductor on which the return signal was detected, the detector 228 can identify the key being pressed.

Once the detector identifies the key being pressed, the processing and communication circuitry 202 preferably generates a key code indicative of the key being pressed. The key code can be a number or other identifier that uniquely identifies a key. The key code can be transmitted to the portal client computer 104 in any suitable manner, as discussed above.

Mapping Module

Figure 2(b) illustrates a mapping module according to an embodiment of the present invention. In the illustrated embodiment, the mapping module 204 includes a card reader 240 for reading a bar codes 242 and 244 on a card 246 insertable into the card reader 240. The card reader 240 includes one or more optical sensors 248 and 250 for reading the bars in the bar code and outputting electrical signals indicative of the code. The optical sensor 248 is positioned to read bar code 242 on the card 246, which includes data, such as a card identification code. Accordingly, the optical sensor 248 is preferably

connected to a data input of the processing and communication circuitry **202**. The optical sensor **250** is positioned to read the bar code **244**, which includes sync bits for synchronizing the reading of the bar code **242**. Accordingly, the optical sensor **250** is preferably connected to a sync input of the processing and communication circuitry **202**.

The present invention is not limited to having the bar code **242** on the top of the card and the bar code **244** containing synchronization bits on the bottom. In an alternative embodiment, the bar code **242** could be on the bottom and the bar code **244** could be on the top. In addition, either or both of the bar codes **242** and **244** can be replaced by a magnetic strip, an eeprom, or any other information-bearing medium.

The bar code **244** includes translucent or non-reflective areas **246** and opaque or reflective areas **248**. When the user inserts a card into the card reader **240**, the translucent or reflective areas and the opaque or reflective areas pass the optical sensor **250** and the bar code **242** passes the optical sensor **248**. Transitions from areas **246** to areas **248** produce rising edges, i.e., a clock signal, on the sync input of the processing and communication circuitry **202**. The rising edges generate interrupts to the processing and communication circuitry **202**. The interrupts cause the processing and communication circuitry **202** to read and process the data bits in the bar code **242**. The bars in the bar code **242** must be correctly aligned with transitions in the bar code **244** so that data bits are read at the correct time.

The processing and communication circuitry **202** preferably executes a software routine for reading the data from the bar code **242**. For example, the

bar code **242** can include a preamble portion, a data portion, and a postamble portion. The processing and communication circuitry **202** uses the preamble portion to detect card insertion. For example, the preamble portion can be a bit sequence, such as "0101". If this sequence is detected as an initial pattern,
5 i.e., when the card reader is activated, the processing and communication circuitry **202** determines that a card is being inserted.

If the initial pattern detected by the processing and communication circuitry **202** is not an initial pattern, the processing and communication circuitry **202** determines whether the initial pattern is the postamble portion in reverse,
10 indicating that a card is being removed from the card reader. For example, if a valid postamble pattern is "0011", and the card reader detects "1100", the processing and communication circuitry **202** determines that a card is being removed.

If the processing and communication circuitry **202** detects any other
15 initial pattern, the processing and communication circuitry **202** determines that the preamble is invalid and takes appropriate action. For example, invalid patterns can cause the processing and communication circuitry **202** to enter a timeout period during which all data is ignored. The processing and communication circuitry **202** can also reset data variables for storing data read
20 from the bar code **242**. After waiting for a predetermined time period, the processing and communication circuitry **202** can return to waiting for a valid preamble or postamble.

Data is only considered valid if the optical sensors scan a valid preamble, e.g., "0101", the correct number of data bits, e.g., 24 bits or 32 bits,

55E260"02095T09

and a valid postamble "0011". When the card is fully inserted, the sync sensor
250 should be blocked. Therefore, a valid insertion event results in preamble,
data, postamble, and sync blocked for extended time period. A valid removal
consists of the reverse data sequence followed by sync path clear. Invalid card
5 activity will trigger a card insertion event with invalid data, such as
"0xFFFFFFFF".

Once the processing and communication circuitry 202 determines that
it has received valid data, the processing and communication circuitry 202
reads the data to determine the card identification code. The processing and
10 communication circuitry 202 then transmits the card identification code, along
with the key being pressed, to the portal client software 102 illustrated in Figure
1. The portal client software 102 uses the card identification code to vary the
association between the keys on the keypad and one or more web servers, as
will be discussed in more detail below.

15 The present invention is not limited to using a bar code and a bar code
reader to vary the association between keys and web servers. For example,
in an alternative embodiment of the present invention, the mapping module 204
can comprise magnetic card reader for reading a magnetic card that encodes
the card identification code. In yet another alternative embodiment, the card
20 can include a memory device and the processing and communication circuitry
202 can be adapted to read the card identification from the memory device. In
yet another alternative embodiment of the invention, the card can comprise a
smart card including a microprocessor and associated memory that can be
used in performing secure e-commerce transactions using the hardware portal

device. In such an embodiment, the mapping module **204** would include appropriate interface circuitry for communicating data between the smart card and the processing and communication circuitry **202**.

5 The present invention is not limited to a mapping module embodied in the portal device **100**. For example, in an alternative embodiment of the invention, the mapping module can comprise a virtual card that can be downloaded to the portal client computer **104** from an external source, such as a friend's website. The virtual card can be read by the portal client software **102** executing on the portal client computer **104** to which the portal device **100** is attached. The portal client software **102** can receive the key code generated by the portal device **100** and use the information on the virtual card to generate the correct query, just as if a physical card had been inserted in the mapping module **204**. For example, if the virtual card is an "ABC News" virtual card and the "NEWS" key is pressed, the portal client software **102** can generate a query to "abcnews.com".

Portal Client Software

Referring back to Figure 1, the portal client software **102** executing on the portal client computer **104** receives user input from the portal device **100** and communicates with the portal server software **106** executing on the portal server computer **108** and with web servers **110**. Figure 3 is a block diagram illustrating exemplary client software according to an embodiment of the present invention. In the illustrated embodiment, the portal client software **102** includes a device driver **300**, a controller **302**, and a browser **304**. The device driver **300** receives signals including key codes and card identification codes

from the portal device 100, converts the signals to data recognizable by the controller 302 and writes the data into a memory location accessible by the controller 302. For example, if the portal device is configured to communicate with the portal client computer 104 using through the USB interface of the portal client computer 104, the device driver 300 can be configured to read a USB data stream and write data from the USB data stream to main memory of the portal client computer 104.

The controller 302 receives the data from the device driver 300, formulates queries and passes the queries to the browser 304. The queries can comprise URL strings or other identifiers for accessing web servers. Passing the query to the browser can be done in a number of ways, for example, using the COM interface, available in Microsoft WINDOWS® operating systems or directly invoking the browser with a command line argument, e.g., "netscape.exe www.abcnews.com".

The controller 302 preferably also directly passes information to the portal server software 102, without using a browser interface. This communication can be achieved using standard networking protocols, such as UDP or TCP, or using HTTP over TCP/IP. The portal client software 102 and the portal server software 106 can exchange information for a variety of reasons, including keeping the client software current, communicating user preference and identification information, with the user's permission, to the portal server software, and for receiving information from the portal server software.

The controller 302 can also manage a portal client database 306 that stores user preferences for one or more users to influence the formulation of queries. The formulation of queries can be based on data received from the portal device 100, the portal server software 106, and/or the portal client database 306.

The browser 304 can be any type of web client adapted to forward queries to external machines and receive data from the external machines. The browser 304 can also provide a graphical user interface for displaying information from the external machines to the user and allow the user to interact with the external machines. Exemplary browsers suitable for use with the present invention include Internet Explorer available from Microsoft Corporation and Netscape Navigator, available from Netscape Corporation.

The present invention is not limited to using a browser a separate controller for formulating queries and communicating with an external machine. For example, the controller 302 and the browser 304 can be integrated within a single software module to communicate with the portal device 100, external machines, the user, and the portal client database 306.

Portal Server Software

Referring back to Figure 1, the system for facilitating user access to websites using a portal device includes portal server software 106 executing on portal server computer 108 for communicating with portal client software 102 executing on portal client computer 104. In the illustrated embodiment, the portal server software 106 includes a server application 400 that reads information from and writes information to a portal server database 402. The

server application 400 can be any type of server capable of communicating with the portal client software 102 and/or the web servers 110 over a computer network. For example, the server application can be configured to communicate using TCP/IP, UDP/IP, or HTTP over TCP/IP. Communication
5 between the portal server application 400 and external devices will be discussed in more detail below.

The portal server database 402 comprises information regarding clients and web server names accessible by the server application 400. For example, the portal server database 402 can store information regarding user
10 preferences, and information for mapping key codes and card codes to web server names. The portal server application 400 is preferably capable of storing information regarding clients in the portal server database 402 and extracting web server address or name information from the portal server database 402 using information received from the portal client software 102.

Communication Methods

15

Figure 5 is an exemplary partial block/partial flow diagram illustrating an exemplary method for facilitating user access to web servers according to an embodiment of the present invention. The blocks have the same general structure and functions as described above, and thus this discussion need not
20 be repeated. The arrows connecting the blocks indicate message flows between the blocks in an exemplary method for facilitating user access to web servers using a portal device. The number beside each arrow indicates the order in which the message occurs. The message flow will be explained in as a series of steps performed by the various devices according to the present

60156020-092309

invention. The steps can be implemented by hardware, software, or a combination of hardware and software.

In step 1, the portal device 100 detects that a key or an area is being pressed or otherwise actuated by the user and formulates a key press package to the portal client software 102. The key press package can include a key code indicative of the key being pressed, a card code indicative of a card inserted in the mapping module, and a device code indicative of the hardware portal device 100. The device code and the card code are optional and need not be sent.

In step 2, the portal client software sends a user identification code, a password, and the key press package to the portal server application 400. The user identification code and the password can be received from the user when the user invokes the portal client software 102. These fields are optional. However, in a preferred embodiment, these fields are sent, in order to allow the portal server application 400 to authenticate the user and collect information regarding the user.

In step 3, the portal server application performs user authentication by verifying that the password is correct for the user identification specified in the request and requests user data from the database. For example, the portal application can perform a table lookup in the database 402 using the key code, the card code, and/or the device code to determine a web server name that the user desires to access. The portal server application 400 can also record user information, such as the web server that the user is accessing. In order to obtain permission from the user to record and sell his or her information, the

portal client software **102** can include an agreement to which the user must agree before the software can be successfully installed. Alternatively, the user can be prompted during authentication of his or her user identification code and password as to whether or not the user consents to the recording and selling of his or her preference information. If the user consents to recording of his or her identification and preference information, the information can be stored in the database **400**. If the user does not consent, the user identification and preference information are preferably not stored in the database **402**.

In step 4, the portal server application **400** forwards the web server name or list of web server names to the portal client software **102**. For example, the web server name or list of web server names can be in HTTP format such that the website corresponding to the web server name or names can be displayed on the browser. If a list of names is communicated to the user, the list can be in the form of an index page that includes a plurality of links to the web servers that the user desires to access.

In step 5, the portal client software **102** transmits a query to the appropriate web server **110** based on the user's selection from the list of web server names or, if a single web server name was returned, based on the information received from the portal server software **102**. In step 6, the selected web server **110** returns a response containing the desired information to the portal client software **102**. For example, the response can contain data for displaying a web page on a display device connected to the portal client computer **104**.

The embodiment illustrated in Figure 5 illustrates exemplary steps by which information entered by the user through a portal device is used to access a web server of interest. All the user is required to do is press a key, and the appropriate information is extracted from the portal server, returned to the portal client application, and used to display the appropriate website to the user.

Figure 6 illustrates an alternative method for facilitating user access to web servers according to an embodiment of the present invention. In Figure 6, the portal client software 102 accesses information in the portal client database 306 to facilitate user access to web servers.

In step 1, the portal device 100 detects that a key or an area has been pressed or otherwise actuated by the user and formulates a key press package to the portal client software 102. The key press package can include a key code indicative of the key being pressed, a card code indicative of a card inserted in the mapping module, and a device identifier that identifies the hardware portal device. The device identifier and the card identifier are optional and need not be sent.

In step 2, the portal client software 102 extracts user profile information from the portal client database 306 and/or updates user profile information in the portal client database 306. The user profile information can include geographic information, such as a postal code, for the user, or user preference information, for facilitating access to web servers. In order to update user profile information in the portal client database 306, the portal client software

can transmit the user identification information, and optionally a password, and the key press package to the portal client database 306.

In step 3, the portal client software 102 transmits the key press package and any user profile information, such as the postal code, to the portal server software 102. As stated above, the transmission can be in TCP/IP format, UDP/IP format, or HTTP over TCP/IP format.

In step 4, the portal server application 400 updates profile information from the client in the database 402 if changes have occurred to the information in the portal client database 306. If the user profile information has not changed, in step 5, the portal server application 400 requests that one of the web servers, as specified by the key press package and any user profile information, transmit information to the portal server application, e.g., using the HTTP redirect command. In step 6, in response to the re-direct command, the accessed web server transmits data to the portal client application for allowing the user to view the desired web page.

Thus, unlike the embodiment illustrated in Figure 5, the embodiment illustrated in Figure 6 does not require that the portal server software communicate the web server name back to the portal client software before the web server can be accessed. The user's request is processed by the portal server software, which directs the appropriate web server to send data to the client. In addition, client information can still be collected in the portal server database, since the original request from the protocol client software is transmitted through the portal server software.

Figure 7 illustrates another embodiment of the present invention for facilitating user access to web servers. Like the embodiment illustrated in Figure 6, the portal client software includes the portal client database 306 for storing user profile information. However, unlike the embodiment illustrated in Figure 6, the portal client software 102 communicates directly with one or more of the web servers 110 to view a desired web page.

In step 1, the portal device 100 detects that a key or an area has been pressed or otherwise actuated by the user and formulates a key press package to the portal client software 102. The key press package can include a key code indicative of the key being pressed, a card code indicative of a card inserted in the mapping module, and a device code indicative of the hardware portal device. The device code and the card code are optional and need not be sent.

In step 2a, the portal client software 102 extracts user profile information from the portal client database 306 and/or updates user profile information in the portal client database 306. The user profile information can include geographic information, such as a postal code, for the user, or user preference information, for facilitating access to web servers. In order to update user profile information in the portal client database 306, the portal client software 102 can transmit the user identification information, an optional password, and the key press package to the portal client database 306.

In step 2b, the portal client software 102 formulates a query to one of the web servers based on at least one of, the key press package, the user profile information extracted from the database 306, the card id, and the device

60156020 092399

id. The query can be in TCP/IP format, UDP/IP format, or in HTTP over TCP/IP format.

Concurrently with step 2b, the portal client software can transmit user profile information to the portal server application 400 so that the portal server application 400 can update user profile information in the database 402. The portal server application can also, in step 2d, sends updates to the user profile information to the client side, e.g., to notify that the client that a web server in the client's personal profile has been moved. Because the portal server application 400 and the portal client software 102 preferably interact to maintain consistency between the databases 306 and 402, the information in both databases is kept current.

In step 3, the web server contacted by the portal client software transmits the requested information to the client.

Thus, the embodiment illustrated in Figure 7 includes the advantages of direct client access to web server, while still allowing updating of user profile information on both client and server sides.

Portal Device Physical Configuration

Figure 8 illustrates an exemplary portal device physical configuration according to an embodiment of the present invention. In the illustrated embodiment, the portal device includes a housing 800 for housing the portal device circuitry. A cable 802 extends from the housing for connecting the internal circuitry to an external computer. In the illustrated embodiment, the cable 802 is a USB cable for connecting to the USB port of the portal client computer. The housing includes a card slot 804 for receiving a card 246.

A keypad **220** is located in open area **806** of the housing **800**. The keypad includes a plurality of keys **222a - 222c**, each including indicium related to web servers that a user may desire to access. For example, keys, such the key **222a**, each have an indicium, such as "jobs", "tickets", or "music" for accessing categories of web servers. Keys, such as the key **222b**, each include an indicium, such as "Dominos", "AT&T", or "EBAY", that indicates company names or logos. Finally, keys, such as keys **222c**, include generic indicia that correspond to specific indicia **808** on the card **246**. In other words, the indicia **808** on the card **246** indicates the function of the indicia on the keys **222c**. The function of the keys **222c** change according to the card **246** inserted in the slot **804**.

Figure 9 illustrates exemplary embodiments of the card **246**. In the illustrated embodiment, cards **246a-246d** each include different indicia **808a-808d** for indicating different functions of the keys **222c**. Cards **246a-246d** also include different company identifiers **810a-810d** indicative of the company accessed by the portal device when the card is inserted in the portal device and the user presses one of the keys **222c**. Although the cards can be any of the types of cards described above, in the illustrated embodiment, the cards include bar codes **242a-242d** for instructing software downstream from the cards **242a-246d** to alter the function of the keys **222c** on the portal device. An exemplary mechanism for altering the function of the keys **222c** is described in detail above and need not be repeated herein. For example, card **810a** is a Barnes & Noble card. When the user inserts the card **810a** in the portal device and presses the key **222c** labeled "home", the portal device causes the

portal client software to access "barnes&noble.com". Similar functionality is provided by the cards 810b-810d. Thus, a user can have a single portal device and a plurality of cards. As a result, the portal device can be easily reconfigured, simply by placing a new card in the card slot, according to the web server or servers that the user desires to access.

Business Methods Using Portal Device

According to another aspect, the present invention includes methods of doing business using portal devices. For example, a company that distributes hardware portal devices can sell key space on the hardware portal device to other companies. When a company buys key space on the hardware portal device, an indicium of the company is placed on one or more keys on the keypad. In addition, the portal client software and/or the portal server software is configured to generate queries to the purchasing company's website or websites when the key or keys owned by the company are pressed. The portal device and the portal client software are then distributed to a plurality of users. Although the portal devices and the portal client software can be sold to the users, in a preferred business method according to the invention, the portal devices and the portal client software are distributed to users without charge. Distributing the portal devices and portal client software to users without charge increase the number of users of portal devices. Accordingly, website hits will increase for companies that own key space on the portal device. Like the portal devices, the cards 246a-246d can be distributed to users with or without charge to the end users to increase hits to company-sponsored web servers.

It will be understood that various details of the invention can be changed

without departing from the scope of the invention. Furthermore, the foregoing description is for the purpose of illustration only, and not for the purpose of limitation--the invention being defined by the claims.

60156020-02095109

CLAIMS

What is claimed is:

1. A method for facilitating user access to the web servers using a portal device, the method comprising:

- 5 (a) receiving, at a portal client computer, a key code generated by a portal device indicative of a key being pressed by a user for accessing a web server;
- (b) transmitting a query over a network for contacting a web server based on the key code;
- 10 (c) storing information pertaining to a user or user preferences in a database; and
- (d) receiving data from the web server based on the query.

2. The method of claim 1 wherein transmitting a query over a network comprises transmitting a query to a portal server computer, the query
15 including the key code.

3. The method of claim 2 comprising receiving a response from the portal server based on the key code, the response including the web server name.

4. The method of claim 3 comprising transmitting a second query to
20 the web server specified by the web server name.

5. The method of claim 2 wherein the portal server computer transmits a redirect message to a web server based on the key code and the web server sends data to the portal client computer.

60156020.092399
662260.02095109

6. The method of claim 1 further comprising accessing a database using the key code to obtain a web server name based on the key code and wherein transmitting a query over a network comprises transmitting a query from the portal client computer to the web server specified by the web server name.

7. The method of claim 1 wherein storing information pertaining to the user or user preferences in a database includes transmitting information pertaining to the user or user preferences to a portal server computer and storing the information in a database managed by the portal server computer.

8. The method of claim 1 wherein storing information pertaining to the user or user preferences in a database includes receiving information pertaining to user preferences from a portal server computer and storing the information pertaining to user preferences in a database managed by the portal client computer.

9. The method of claim 1 wherein storing information pertaining to the user or user preferences in a database includes storing the information in first database managed by the portal client computer and in a second database managed by a portal server computer connected to the portal client computer over a network.

60156020 092399

10. A computer program product comprising computer-executable instructions embodied in a computer-readable medium for performing steps comprising:

- 5 (a) receiving, at portal client computer, a key code generated by a portal device indicative of a key being pressed by a user for accessing a web server;
- (b) transmitting a query over a network for contacting a web server based on the key code;
- 10 (c) storing information pertaining to a user or user preferences in a database; and
- (d) receiving data from the web server based on the query.

11. The computer program product of claim 10 wherein transmitting a query over a network comprises transmitting a query to a portal server computer, the query including the key code.

15 12. The computer program product of claim 11 comprising receiving a response from the portal server based on the key code, the response including the web server name.

20 13. The computer program product of claim 12 comprising transmitting a second query to the web server specified by the web server name.

14. The computer program product of claim 10 wherein storing information pertaining to the user or user preferences in a database includes transmitting information pertaining to the user or user preferences to a portal

662250-02095709

server computer and storing the information in a database managed by the portal server computer.

15. The computer program product of claim 10 wherein storing information pertaining to the user or user preferences in a database includes
5 receiving information pertaining to user preferences from a portal server computer and storing the information pertaining to user preferences in a database managed by the portal client computer.

16. The computer program product of claim 10 wherein storing information pertaining to the user or user preferences in a database includes
10 storing the information in first database managed by the portal client computer and transmitting the information to a portal server computer for storage in a second database managed by the portal server computer.

17. A portal device for facilitating user access to web servers comprising:

- 15 (a) a user interface having a plurality of keys or areas for receiving user input regarding a web server or web servers that a user desires to contact, each key being associated with a web server or a plurality of web servers;
- 20 (b) processing and communication circuitry operatively associated with the keys or areas for identifying a key or area being actuated by a user, producing a key code indicative of the key or area, and forwarding the key code to a computer; and

- (c) a mapping module for producing information for varying the association between at least some of the keys or areas and web servers.

5 18. The portal device of claim 17 wherein the mapping module comprises a card reader coupled to the processing and communication circuitry for reading codes from cards insertable in the card reader, wherein the codes are used for varying the association between the keys or areas and web servers.

10 19. The portal device of claim 18 wherein the card reader comprises a bar code reader for reading bar codes containing information for varying the association between at least some of the keys or areas and web servers.

15 20. The portal device of claim 18 wherein the card reader comprises a magnetic card reader for reading magnetically encoded information for varying association between at least some of the keys or areas and web servers.

21. The portal device of claim 18 wherein the card reader comprises a memory card reader for reading information stored in a memory card for varying the association between at least some of the keys or areas and web servers.

20 22. The portal device of claim 17 wherein the mapping module comprises a smart card interface for receiving information from a smart card for varying the association between at least some of the keys or areas and web servers.

663260:0205109

23. The portal device of claim 17 wherein the mapping module comprises a virtual card downloadable from a web server for varying the association between at least some of the keys or areas and web servers.

24. A method for generating revenue and for increasing hits to
5 company sponsored websites, the method comprising:

- (a) selling key space on portal devices to companies, the key space including a key or an area having a company or product identifier and being adapted to producing a code for generating a query to a company-sponsored web server; and
- 10 (b) distributing the portal devices to users to increase access to the company-sponsored web servers indicated on the portal devices.

25. The method of claim 24 wherein distributing the portal devices to users includes distributing the portal devices to the users without charge to
15 users.

26. The method of claim 24 wherein distributing the portal devices to users includes selling the hardware portal devices to users.

27. The method of claim 24 comprising recording user identification information and user preference information generated by the portal devices.

20 28. The method of claim 27 comprising selling the user identification and preference information.

29. The method of claim 24 comprising distributing cards to users having company or product identifiers, wherein the cards are insertable in the

50156020.092399
55E260-02095109

portal devices for modifying the portal devices to access web servers relating to the company or product identifiers on the cards.

30. The method of claim 24 wherein distributing cards to users comprises distributing cards to users without charge to users.

5 31. The method of claim 24 wherein distributing cards to users comprises selling the cards to users.

66E260-02095T09

Abstract of the Disclosure

Methods and systems for facilitating user access to web servers include a portal device that includes keys or areas associated with web servers. Processing and control circuitry detects when a user presses one of the keys or areas and forwards a key code to a portal client computer. The portal client computer sends a query for accessing a web server based on the key code.

5

662260-02095109

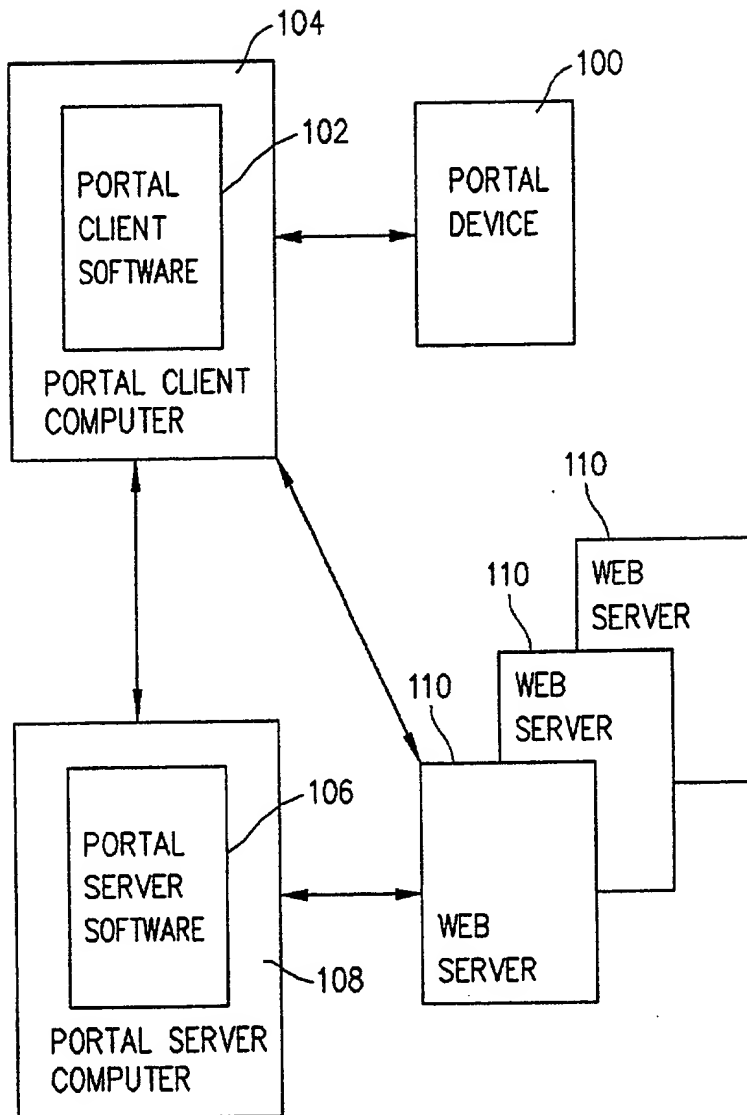


FIG.1

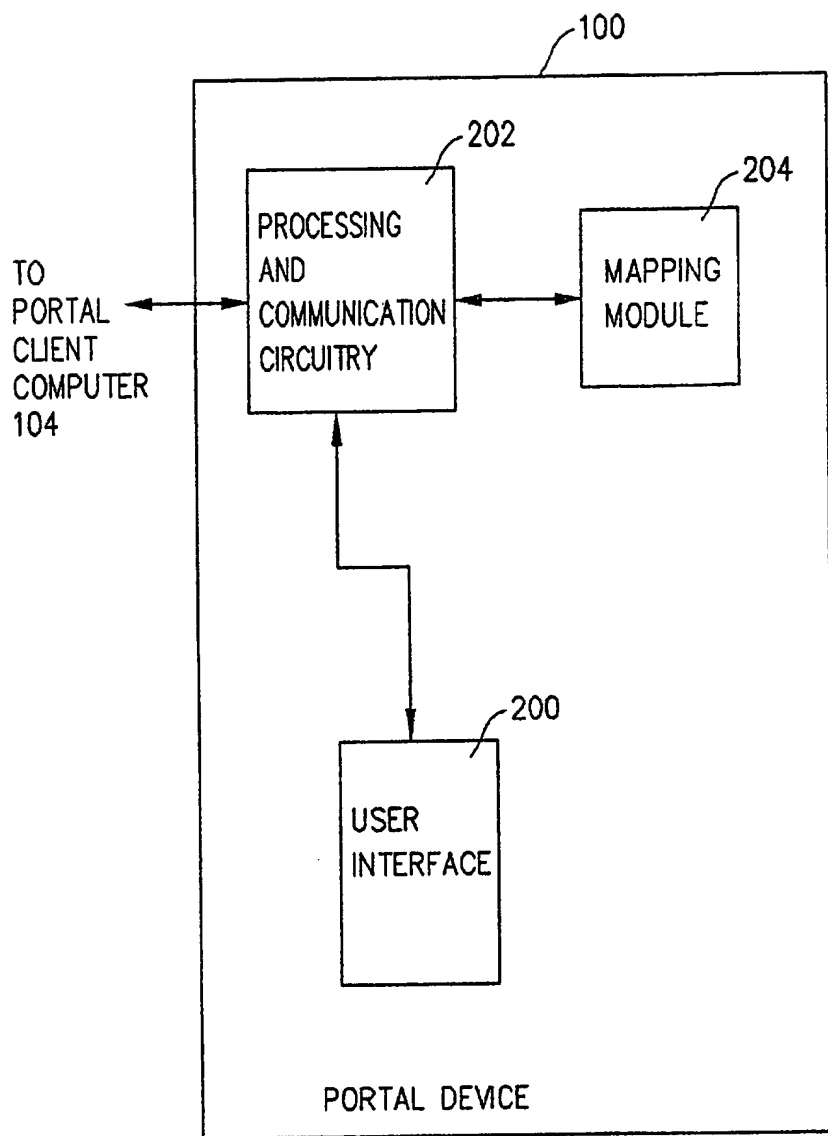


FIG.2

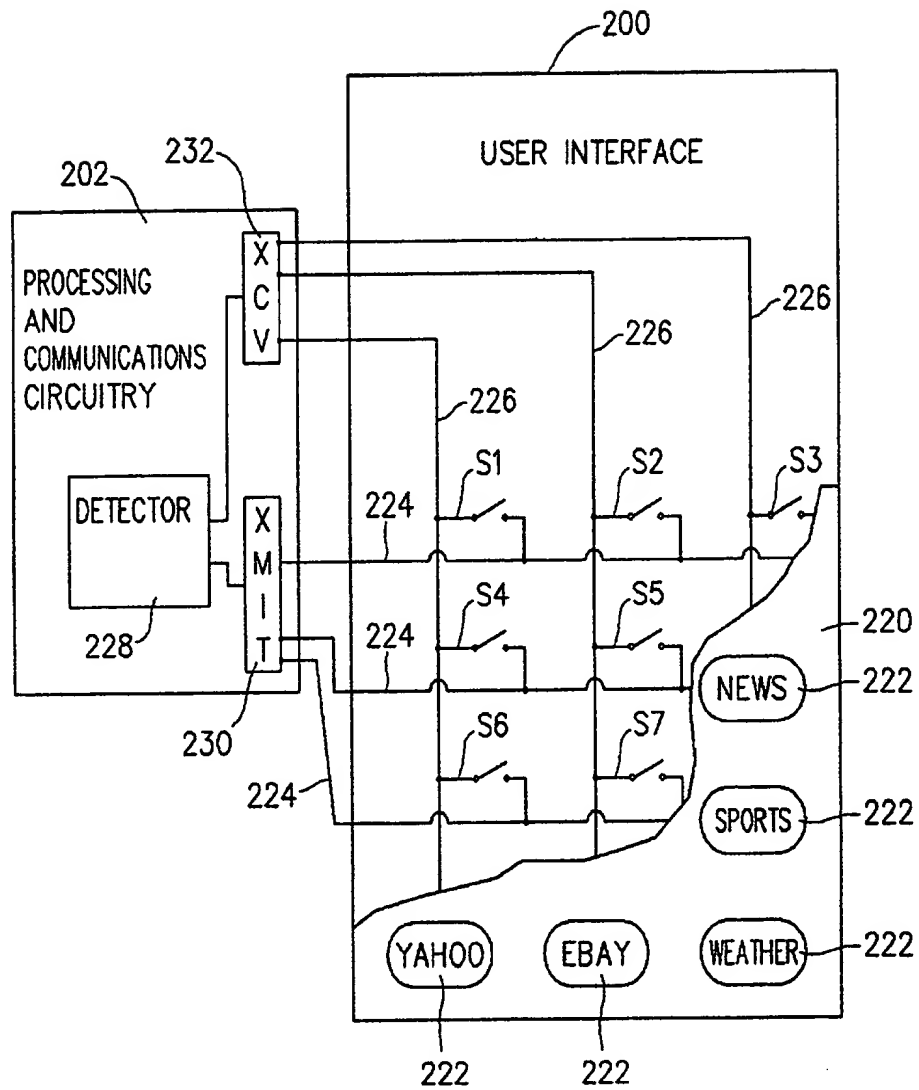


FIG.2(a)

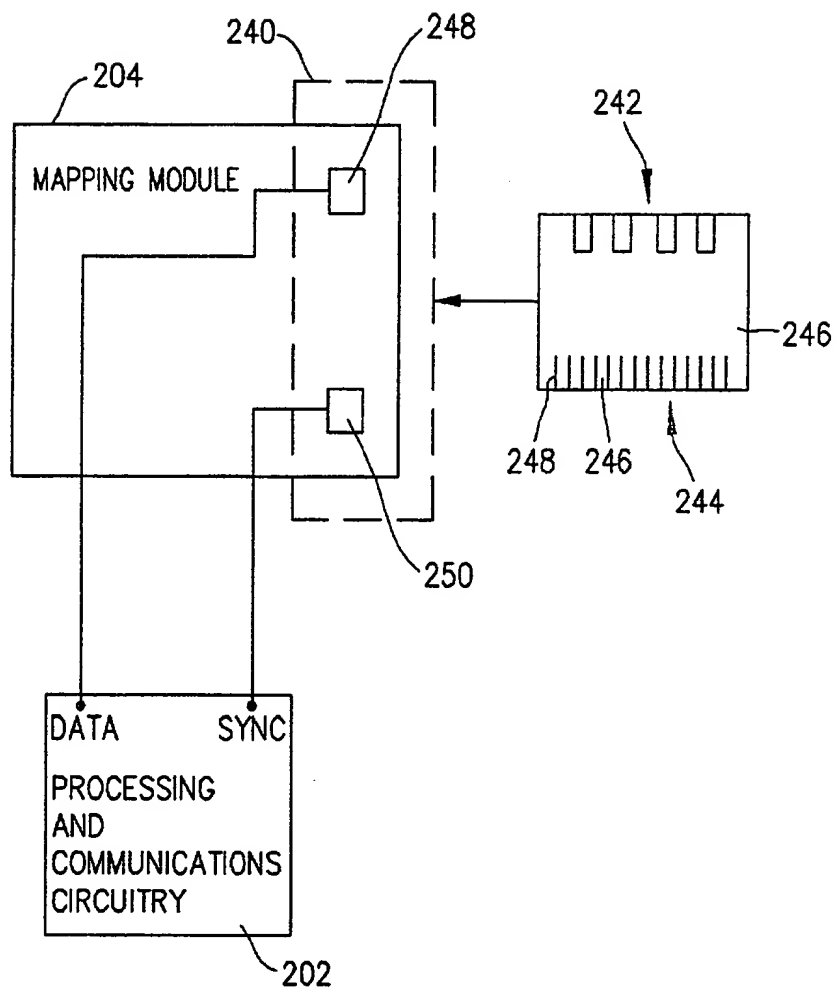


FIG.2(b)

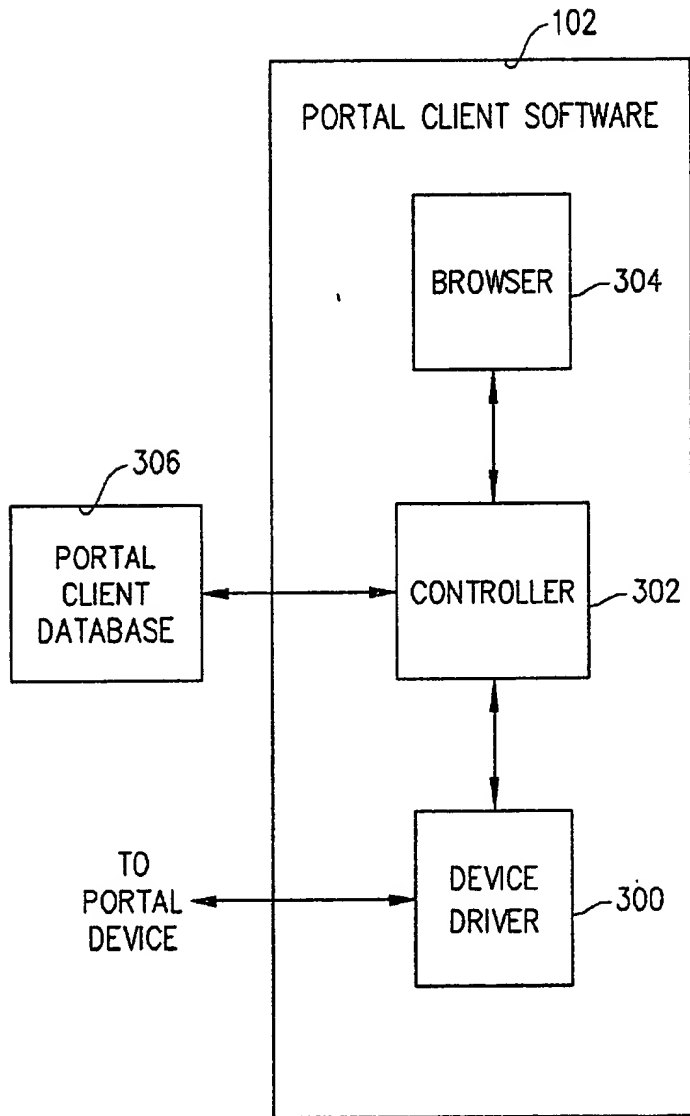


FIG.3

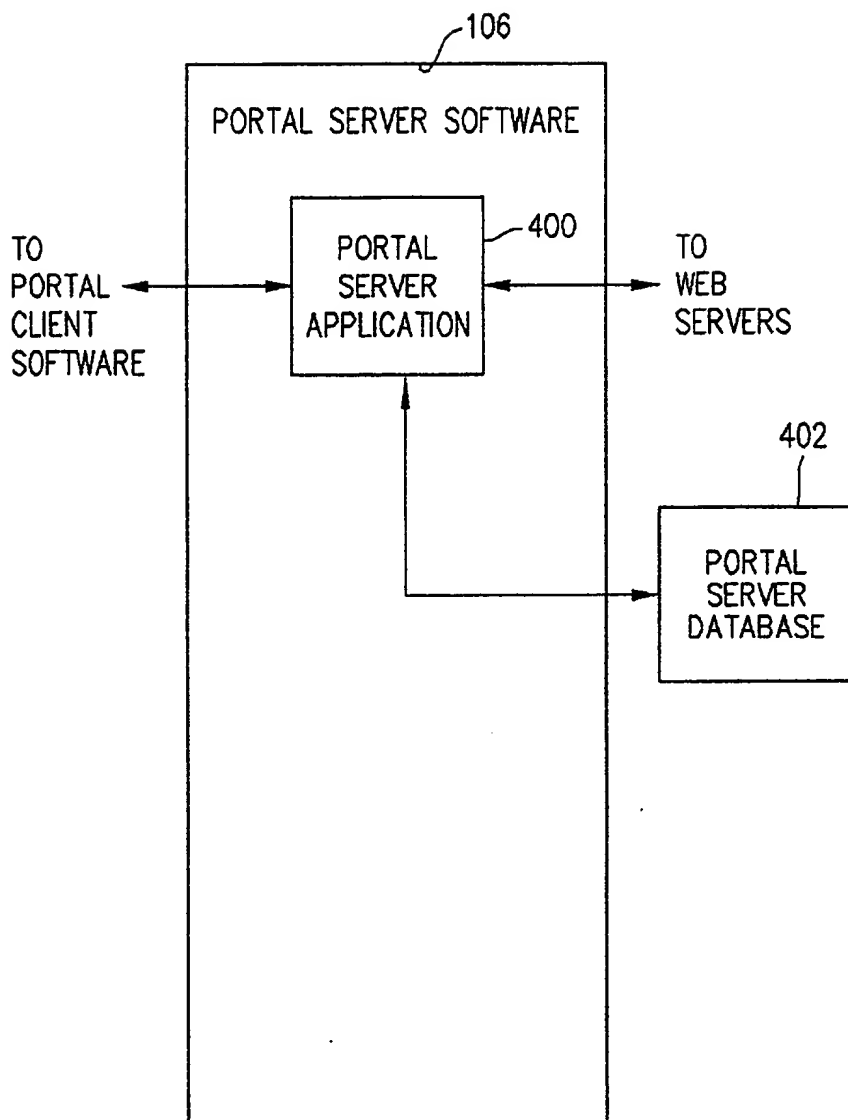


FIG.4

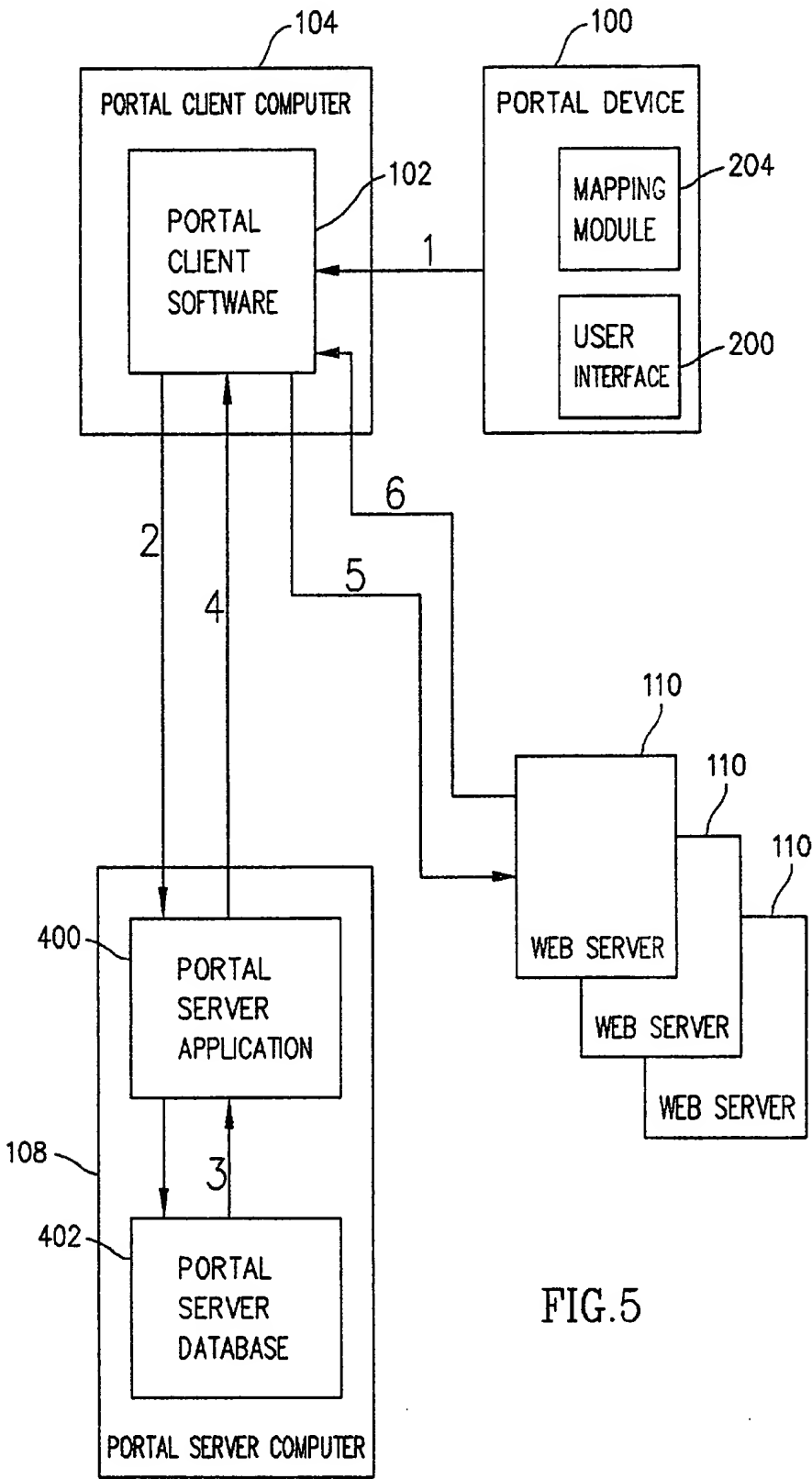


FIG.5

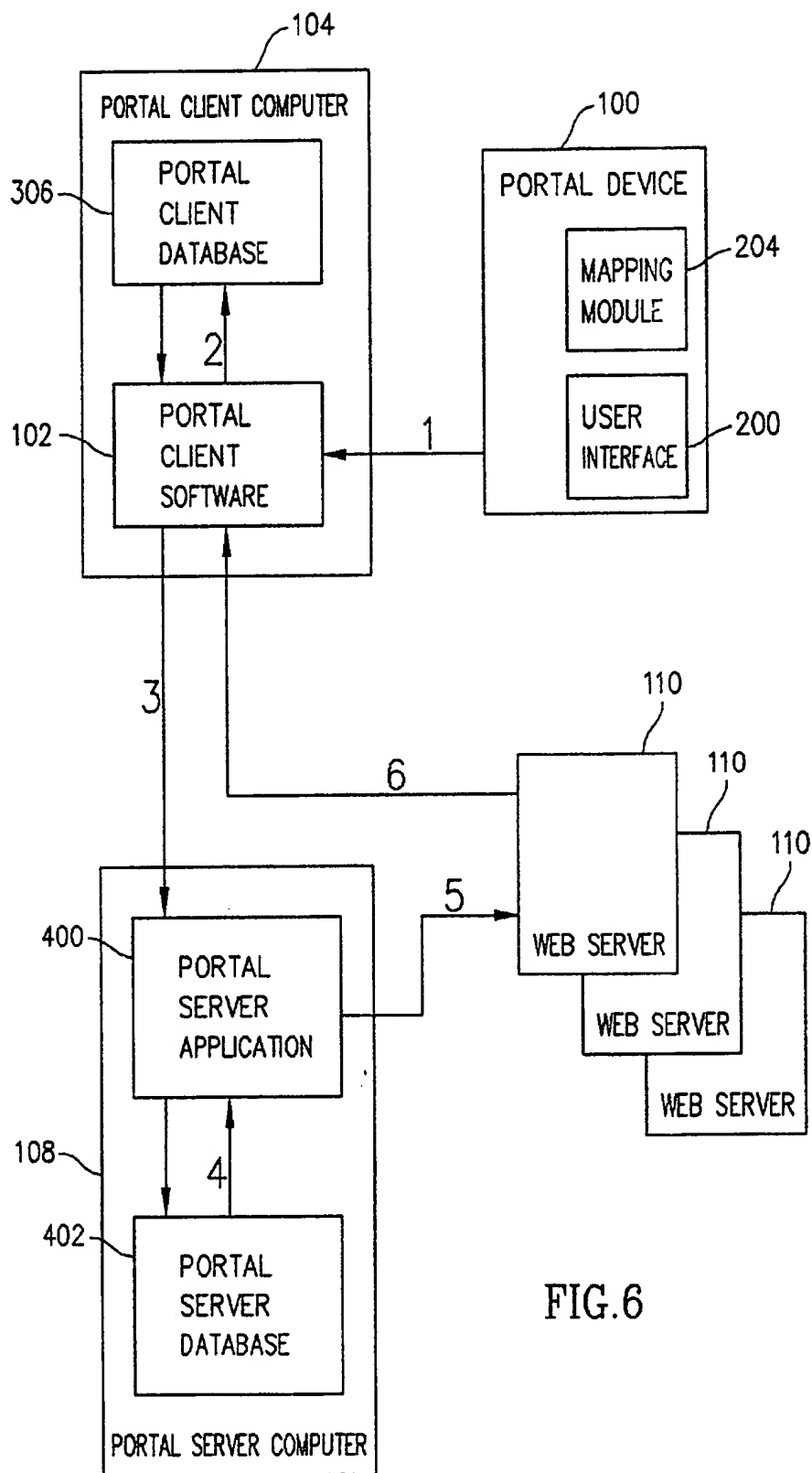


FIG.6

FIG. 7

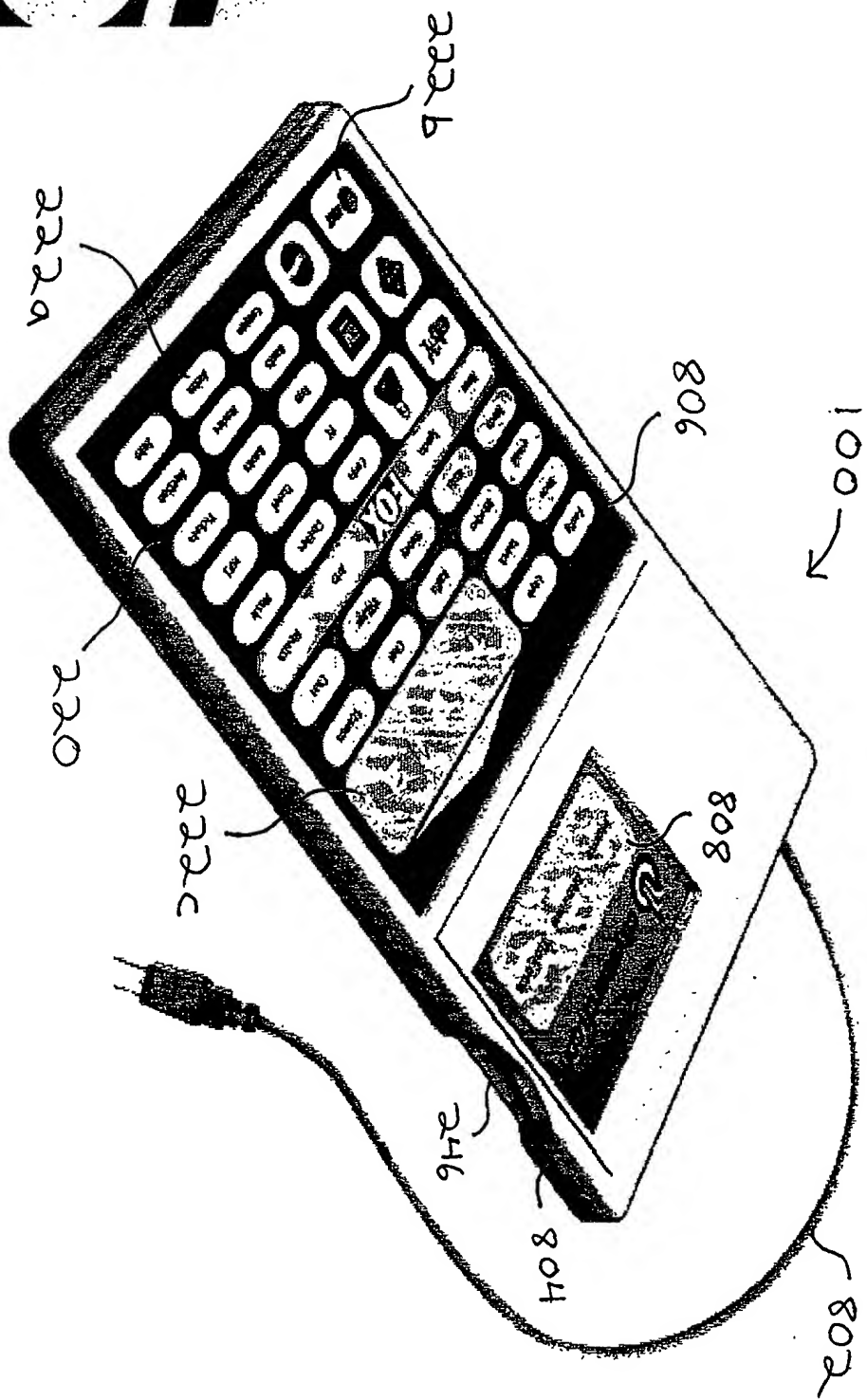


FIG. 8

FIG. 8, 092399

SlideCard™ Examples

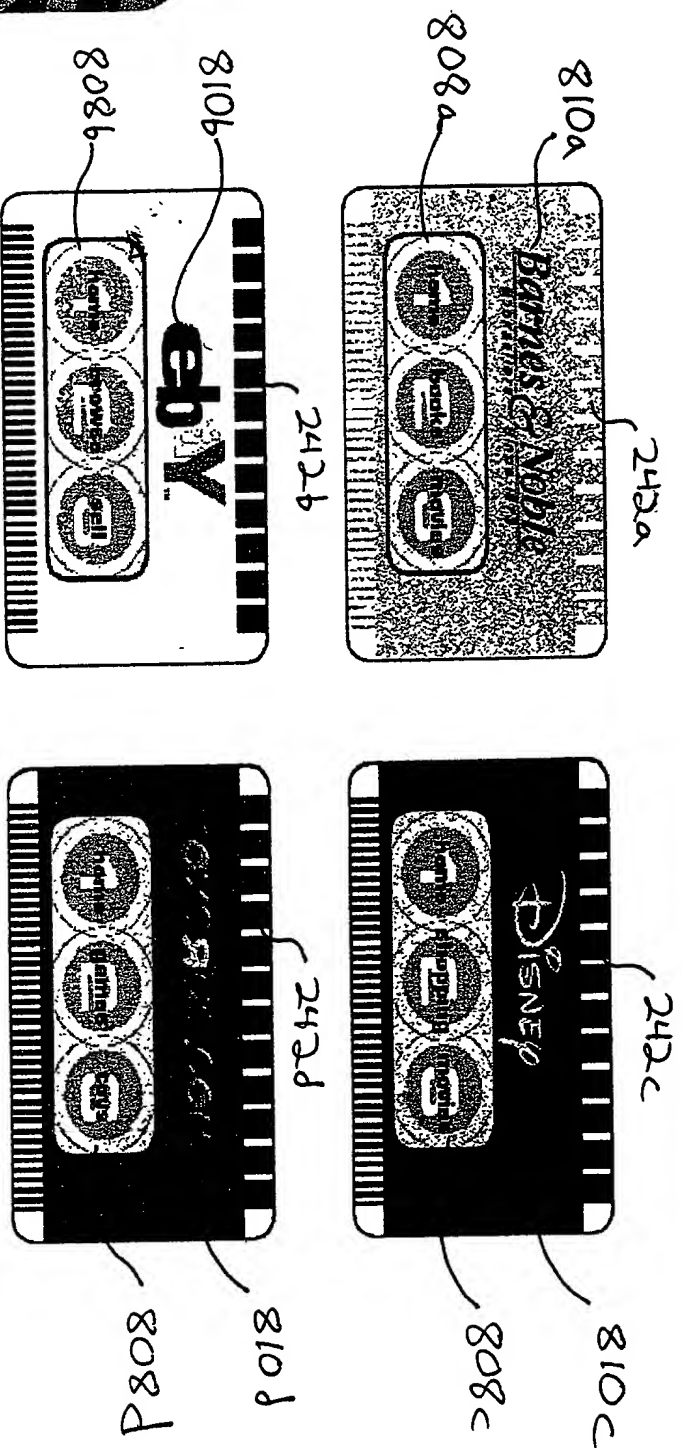


FIG. 9

50156020.092399

02/04/00

02/04/00

J490 U.S. PTO

Please type a plus sign (+) inside this box → ☒

Lynette M. Bailey

PTO/SB/16 (2-98)

Approved for use through 01/31/2001 OMB0651-0037
Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

JCS53 U.S. PTO
60/180466
02/04/00

INVENTOR(S)					
Given Name (first and middle [if any])		Family Name or Surname		Residence (City and either State or Foreign Country)	
David Louis		Kaminsky		Chapel Hill, North Carolina	
Brent M		Kleinheksel		Durham, North Carolina	
David Mark		Ogle		Cary, North Carolina	
<input checked="" type="checkbox"/> Additional inventors are being named on the <u>1</u> separately numbered sheets attached hereto					
TITLE OF THE INVENTION (280 characters max)					
IMPROVED PROCESS FOR HANDLING AND EXPLOITING SLIDE CARD USE					
<div> <div>Direct all correspondence to:</div> <div> <input type="checkbox"/> Customer Number <input type="text"/> → <div>Place Customer Number Bar Code Label here</div> </div> <div>OR</div> <div> <input checked="" type="checkbox"/> Firm or Individual Name <u>Jeffrey L. Wilson</u> </div> </div>					
Address		<u>Suite 1400 University Tower</u>			
Address		<u>3100 Tower Boulevard</u>			
City	<u>Durham</u>	State	<u>NC</u>	ZIP	<u>27707</u>
Country	<u>USA</u>	Telephone	<u>919-493-8000</u>	Fax	<u>919-419-0383</u>
ENCLOSED APPLICATION PARTS (check all that apply)					
<input checked="" type="checkbox"/>	Specification Number of Pages	<u>37</u>	<input checked="" type="checkbox"/>	Small Entity Statement (Unexecuted)	
<input checked="" type="checkbox"/>	Drawing(s) Number of Sheets	<u>9 Figures (8 Sheets)</u>	<input checked="" type="checkbox"/>	Other (specify) <u>Express Mail # EJ227696660US</u>	
METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT (check one)					
<input checked="" type="checkbox"/>	A check or money order is enclosed to cover the filing fees				FILING FEE AMOUNT (\$)
<input type="checkbox"/>	The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number. <input type="text"/>				<u>75.00</u>
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.					
<input checked="" type="checkbox"/> No.					
<input type="checkbox"/> Yes, the name of the U.S. Government agency and the Government contract number are: <input type="text"/>					

Respectfully submitted,

SIGNATURE Jeffrey L. WilsonTYPED or PRINTED NAME Jeffrey L. WilsonTELEPHONE 919-493-8000Date 2, 4, 00REGISTRATION NO. 36,058

(if appropriate)

Docket Number: 1335/9/2**USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT**

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, D.C., 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C., 20231

This
prov.
60/180,466filed
2/4/00Does
notcontain
cited
portions

Kaminsky

+

Approved for use through 01/31/2001. OMB 0651-0037

Patent and Trademark Office, U. S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Docket Number		1335/9/2	Type a plus sign (+) inside this box →	+
INVENTOR(S)/APPLICANT(S)				
Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)		
Gregory Reasoner	DeKoenigsberg	Zebulon, North Carolina		
Thomas Owings	Rowe	Chapel Hill, North Carolina		
Daniel Paul	Gajewski	East Aurora, New York		

Number 1 of 1

+

[illegible]

1. Introduction

1. A U.S. provisional patent application for IMPROVED PROCESS FOR HANDLING AND EXPLOITING SLIDE CARD USE (37 pages);
2. Nine (9) Figures (8 Sheets);
3. Provisional Application for Patent Cover Sheet (Form PTO/SB/16, 2 pages);
4. An unexecuted Small Entity Statement;
5. A check in the amount of \$75.00 to cover the small entity provisional application filing fee;
6. A filing receipt card to be returned to our offices with the U.S. Patent and Trademark Office date stamp thereon; and

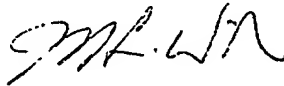
Assistant Commissioner for Patents
February 4, 2000
Page 2

7. A Certificate of Express Mail No.: EJ227696660US.

Please contact our offices if there are any questions.

Respectfully submitted,

JENKINS & WILSON, P.A.

A handwritten signature in black ink, appearing to read "JLW", with a stylized flourish at the end.

Jeffrey L. Wilson
Registration No. 36,058

JLW/GAH/lmb

Enclosures

004020 " 09403709

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Kaminsky et al.

Application No.:

Filed on: Herewith

Title: IMPROVED PROCESS FOR HANDLING AND EXPLOITING SLIDE CARD USE

**STATEMENT CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(b))--SMALL BUSINESS CONCERN**

I hereby state that I am an official of the small business concern empowered to act on behalf of the concern identified below:

PlanetPortal.com, Inc.
2222 Highway 54, Suite 180
Durham, North Carolina 27713

I hereby state that the above identified small business concern qualifies as a small business concern, as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office under Sections 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third-party or parties controls or has the power to control both.

I hereby state that rights under contract or law have been conveyed to, and remain with, the small business concern identified above, with regard to the invention described in the specification filed herewith, with title as listed above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights in the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c), if that person made the invention, or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each such person, concern or organization having any rights in the invention is listed below:

No such person, concern, or organization exists.

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small business entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed..

Brent M. Kleinheksel
Chief Executive Officer
PlanetPortal.com, Inc.
2222 Highway 54, Suite 180
Durham, North Carolina 27713

SIGNATURE: _____ Date: _____

00130466 020406

IMPROVED PROCESS FOR HANDLING AND EXPLOITING SLIDECARD
USE

AN APPLICATION FOR
UNITED STATES LETTERS PATENT

By

David Louis Kaminsky
Chapel Hill, North Carolina

Brent M. Kleinheksel
Durham, North Carolina

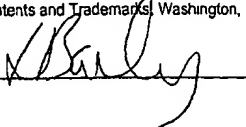
David Mark Ogle
Cary, North Carolina

Gregory Reasoner DeKoenigsberg
Zebulon, North Carolina

Thomas Owings Rowe
Chapel Hill, North Carolina

Daniel Paul Gajewski
East Aurora, New York

004030 094030 004030

"Express Mail" mailing number EJ22769660US
Date of Deposit February 4, 2000
I hereby certify that this paper or fee is being deposited
with the United States Postal Service "Express Mail Post
Office to Addressee" service under 37 C.F.R. 1.10 on the
date indicated above and is addressed to the
Commissioner of Patents and Trademarks, Washington,
D.C. 20231
Lynette M. Bailey 

Description

IMPROVED PROCESS FOR HANDLING AND EXPLOITING SLIDECARD

USE

5

Technical Field

The present invention relates to an improved process for handling and exploiting slide card use.

Background Art

10 Many have observed that it is advantageous to marry the physical world of commerce with the on-line, web world. In one family of scheme, consumer goods, coupons and similar items come printed with unique codes. When a user scans these codes at his computer, a set of processes loads an associate web page into his browser. Such schemes allow
15 consumers easy access to information about products, which creates an advertising opportunity for the seller.

U.S. Patent Number 5,976,773, the disclosure of which is incorporated herein by reference in its entirety, describes a method in which UPC codes attached to items can be scanned with UPC reader attached to a
20 computer, and the code transmitted to a database. The database translates the code into a URL stored in the database for the product, and the web page associated with the URL is loaded into a browser.

004020 "030400

Commonly assigned, copending U.S. Patent Application Number 09/440,318, filed November 12, 1999, the disclosure of which is incorporated herein by reference in its entirety introduces the notion of "buttons" typically present on the reader, and associated with each card. Rather than taking
5 users to a single site, cards can take users to a plurality of sites, each site associated with a button.

However, the current art suffers a number of limitations. First, as described in '773 Patent, the process of translating card numbers into URLs requires a back-end database, situated behind a request-handling web
10 server, as shown in Figure 1 of the '773 Patent. Deploying and maintaining a database is costly, error-prone and hinders scalability.

Second, should adoption grow, the number of cards could get quite large. For example, a 40-bit code can support over 1 trillion unique codes. Deploying a back-end infrastructure that can respond to such a large number
15 of unique queries is expensive and slow.

Third, in the current art, product codes are uniquely mapped to URLs. While this mapping is simple, it does not exploit the demographic diversity advertisers customarily leverage to transmit their message effectively.

Fourth, for cards to map to URL, so a standard for code issuance
20 must exist. If such a standard does not exist, two firms might arbitrarily select the same code. In this case, database queries for the codes could not resolve uniquely, as two firms will register URLs for the same code. The current art suggests no mechanism for automating the process of assigning codes, and printing cards.

004030 "004030"

numbers" without limitation. When the reader is activated on the printed code, it transmits the card number. Such readers are known in the art, and are examples are given in <both patent #s>. Optionally The reader can optionally include one or more buttons. If so, when pressed, the reader
5 transmits an indicium of the pressed button, along with card number from the last card read. We will call the button indicium a "button number" without limitation.

The reader can be a standalone component connected via cable or wireless connection to a computing device. Examples of computing devices
10 include PCs, PDAs, WebTV-type devices, cellular phones, etc.

Request Generator

The Request Generator is responsible for formulating the web query such that it can be handled by a standard web server. The Request
15 Generator takes the card number and optional button number, and formulates a web request. In the preferred embodiment, this request is a URL. The URL has the form: <protocol><server>[<routing>]<card and button indicators> In most web cases, the protocol will be HyperText Transfer Protocol, so the <protocol> will typically be:

20 http:

The server defines the web server that will handle the request. In the simplest case, a well-known server such as

www.planetportal.net

handles all such requests. Below, we will describe an extended mechanism
25 that uses multiple servers to extend scalability.

cards

12/1.html

12/3.html

Thus, for example, when card 12 is swiped, and button 1 is pressed,

<http://www.planetportal.com/cards/12/1.html>

20 be used, and a swipe query for card 12 would be:

The request generator will typically reside in the reader or in the printing device, but can reside in a connected, standalone component.

when passed a URL such as the one above will activate the default browser, and cause it to load the page specified by the URL.

15

When the request reaches the web server, the server uses standard web server technology to fetch the requested page. In the typical case, the web server would return the contents of the page specified by the request.

20 In our example where button 1 was pressed on card 12, the server at www.planetportal.net would look in the "cards" directory, find the "12" directory, then return the contents of the file "1.html". (Note that web servers can be configured to map "virtual directory names" such as "cards" to physical directories on the machine. This allows the server to export one

25 name space externally, while maintaining a different one internally, allowing

the systems administrator more flexibility. For example, "cards" might be translated by the web server into "CardDir", which would be used for the lookup. Such translations are well-known in the art, and do not materially effect this invention.)

5 In one embodiment of this invention, the file on the web server contains a copy of the requested page. For example, if card 12 were issued for Pepsi Cola (TM), the page might contain information on Pepsi. However, if the web server maintained information about all products locally, it would have to store enormous quantities of data.

10 Consequently, in the preferred embodiment, the files such as "12/1.html" contain web redirect indications such as:

<META HTTP-EQUIV="refresh" CONTENT="0;URL=http://www.pepsi.com">

 This indication is transferred back to the user's browser instead of the information. The "refresh" clause tells the browser to load the page from a
15 new location, in this case, <http://www.pepsi.com>.

 When such redirect indications are received by a standard browser, rather than rendering the content, the browser execute a subsequent web query to load the URL indicated in the redirect. Redirect technology is well-known to those skilled in web technology.

20 In an alternate embodiment, the web server can look up the URL specified by the redirect indication, and return the correct page directly to the browser. This solution is not preferred since the web server will receive many requests, so it is advantageous to reduce the amount of processing done by it.

supplement web server function. By eliminating the database, we decrease processing time, and increase scalability.

In addition, using a web server simplifies the process of data collection by exploiting logging mechanisms inherent in many modern web
5 servers. This further reduces complexity. This process is discussed further below.

Further Improving Scalability Via Intelligent Client-based Server Selection

In the scenario described above, the client creates a URL specific to a
10 given card/button pair, and sends it to a predefined server. However, as the number of card transactions grows large, a single server will have difficulty handling them all.

In the current art, multiple servers can be deployed behind a machine that load balances among the servers. This configuration is shown in figure
15 2.

The drawback to this approach is that each server must have access to all web pages that are accessible from any of the servers. When one applies this technique to the case of millions or billions of cards, the resulting amount of information sharing among the servers (typically accomplished by
20 file sharing) becomes overwhelming. Either the file system will prove a bottleneck, resulting in reduced scalability, or each machine needs its own copy of all of the files, resulting in higher costs.

To overcome this limitation, our invention supplement database-elimination with optional the optional use of client-side server selection. In
25 intelligent client-side server selection, the request generator is presented

with a list of servers, and algorithmically maps the card number to a unique server.

In the preferred embodiment, the request generator contains a list of servers that handle card request. Such a list is contained in a simple text

5 file, and is of a form such as:

www1.planetportal.net

www2.planetportal.net

...

wwwN.planetportal.net

10 or an equivalent form. Note that the servers needn't look as similar as they do in the example. The list could be:

www.myfirstserver.com

www.mysecondserver.net

w3.yetanotherserver.edu

15 This list simply defines which servers handle requests.

As illustrated in figure 3, when a request arrives at the Request Generator, the Request Generator determines which server is assigned the card. In the preferred embodiment, the Request Generator uses a simple modulus function. The servers are assigned zero-based indices, and the

20 card number is taken modulus the number of servers. The result is used to select the proper server. For example, if the servers are:

www1.planetportal.net

www2.planetportal.net

www3.planetportal.net

Then there are three total servers, so card numbers are taken modulo three. So, for example, card 1 would be mapped to server 1; card 12 would be mapped to server 0; and card 14 would be mapped to server 2. Using the zero-based indexing, server 0 is www1.planetportal.net; server 1 is
5 www2.planetportal.net; and server 2 is www3.planetportal.net. The computed server name is used in the
<server> field of the URL as defined above.

Note that in the degenerate case, there is a single server, such as:

www.planetportal.net

10 and this method reduces to the simple method described above.

In an alternate embodiment, instead of using lists of servers, the client is supplied simply with the number of servers, and computes algorithmically the server name. For example, if the servers are assigned names such as:

www0.planetportal.net

15 www1.planetportal.net

www2.planetportal.net

www3.planetportal.net

www4.planetportal.net

Then the Request Generator would simply be told that there are five servers.

20 The Request Generator would compute the modulo function, and compute the server name. In this case, if M is the modulus, then the server name is:

wwwM.planetportal.net

Note that more complex functions can be used for mapping card numbers to server names. For example, certain bits in the card number can
25 be combined using boolean algebra to determine the name. Similarly,

simple mechanisms such as downloading a list of mappings of cards to servers can be employed. However, such a file will likely grow large, so algorithmic approaches are preferable.

In some cases, a client will accidentally route requests to an incorrect server. This can happen when a client has an outdated server table. For example, when a new server is added, some portion of the card requests will be assigned to the new server. Until a client gets an indication of this new assignment, some fraction of the requests will be misrouted.

Many modern web servers, including the Apache web server produced by the Apache Group (www.apache.org), allow web masters to specify files that will be returned should a request fail. These files can be specified per requested directory. In the case of a card misrouting, the error-file executes the server-lookup algorithm to determine the proper server, and returns a redirect indicator to the client. The redirect process is described elsewhere in this document. An example of such a script is included below:

```

#!/usr/bin/perl

sub lookup {
    # contains the server lookup algorithm. This is a very
    # basic sample lookup algorithm, using mod 10 to pick a
    # server name from www0 to www9
    my $cardid = $_[0];
    my $serverid = cardid % 10;
    return "www$serverid.planetportal.net";
}

$document_name = $ENV{'HTTP_REFERER'};    # in this implementation,
referer                                     # would contain the
                                           # original
                                           # card filename requested,
although
                                           # this could come from
other sources                             # in other implementations

$document_name =~ s/http(.*?)net//;        # substitute out the bad
server name

```

```

$document_name =~ /(.*)(\d*)\.html/;      # match the numeric
element of the                             # filename and assign it
$card_number = $2;                          # card_number, match the
5  to                                       # and assign to path_data

$path_name = $1;

10  $new_server = &lookup($card_number);    # &lookup function
contains the                               # algorithm, which returns
the                                         # text string containing
15  the                                     # proper server name

$new_document_name = $new_server . $path_name . $card_number .
'.html';
20  # build new URI with
corrected                                  # server name

print "Location: $new_document_name\n\n";

```

25 To handle this case, server re-routing is employed. When a server receives a request for a card that it does not handle, it will attempt to look up the file (or program) associated with the card. This search will fail.

In yet another embodiment, the location of the client is used to determine where the server farm resides. In this case, the Request

30 Generator uses a location indicator along with the card number to find the server. For example, during setup, most PCs ask the user to supply the country in which the PC will be used. This information can be used to append a county code to the URL.

In one embodiment of this scheme, the Request Generator is supplied

35 with a mapping of countries to local servers. For example, China, Japan and Korea might all map to servers in Japan, whose country code is ".jp". Thus, to the computed server name described above, the Request Generator would append ".jp". This scheme further balanced load among servers, and

provided improved response time to card requests by using servers that are geographically nearby.

Such geographical mapping is optional addition to the client-side selection mechanism.

5 Note that none of these schemes preclude the use of server-side load-balancing. Multiple servers can still be deployed to handle any subset of card requests. This is illustrated in figure 4.

 Note further that the files used by the Request Generator to configure the client-side server selection can be downloaded dynamically using the
10 mechanism disclosed in copending application 09/440,318 referenced above.

 In summary, the client-side server selection reduces the load on any given server, and additionally reduce the number of files that need be accessed by any given server. The objective is accomplished by algorithmic
15 assignment of card numbers to server, optionally supplemented by geographic distribution techniques.

Tailored Responses to Card Requests

 The mechanisms described above improve the scalability of the
20 system. However, they do not attempt to improve the user experience, which limits the value to both users and to advertisers. This aspect of the invention improves user experience by tailoring responses to card requests based on knowledge about the user.

Gathering Descriptive Information

To tailor responses to card requests, information about the user must be gathered. This is routinely done by commercial software and by web sites such as "http://my.yahoo.com". Such information includes

5 demographics that can include, but is not limited to: name, address, zip code, e-mail address, gender, income, etc. The mechanisms described below will use such demographic information to select among one more web pages associated with any given card.

10 Selection Using Demographic Information

In the first embodiment, descriptive information is used to further select among files stored on a web server. Descriptive information includes both static information such as gender, address, etc., and behavioral information, such as "user swiped this card ten time."

15 In one simple example, gender can be used to tailor card responses. Recalling the URL format described above:

<protocol><server>[<routing>]<card and button indicators>

We refine this format to be:

<protocol><server>[<routing>]<card and button

20 indicators><demographic indicator>

Thus, the demographic indicator is used to select among multiple files associated with each card/button indicator.

Refining the example given above: when card 12 is swiped, and button 1 is pressed by a male, the query formulated by the Request

25 Generator might look like:

<http://www.planetportal.com/cards/12/1/male.html>.

This file would contain a URL tailored for a male interested in the product advertised by the card.

In cases where the Request Generator created a request, and no
5 demographic specific files are supplied by the server, a web-server failover
technique can be used to insure that the user is supplied meaningful data.
As described above, many modern web servers, including the Apache web
server produced by the Apache Group (www.apache.org), allow web
masters to specify files that will be returned should a request fail. These files
10 can be specified per requested directory. To handle the case above, the
proper redirect file would be used as the failover

file assigned to:

<http://www.planetportal.com/cards/12/1>

That is, all users who request any file in:

15 <http://www.planetportal.com/cards/12/1>,

including

<http://www.planetportal.com/cards/12/1/male.html>, and

<http://www.planetportal.com/cards/12/1/female.html>

would be receive the redirect file associated with card 12, button 1.

20 Note that in some cases, the order of the <routing>, <card and button
indicators>, and <demographic indicator> can be interchanged in some
cases without materially affecting the invention. The fields in the example
above can be transposed to be:

<http://www.planetportal.com/cards/12/male/1.html>.

25 or

004030 "020400

<http://www.planetportal.com/cards/male/12/1.html>.

In all of these cases, the same file would be served.

Using the technique described above, other individual demographic fields can be used, and multiple fields can be combined at the expense of

5 complexity in the file-system structure.

This complexity can be saved at the expense of server-side processing. In this embodiment, the demographics are passed as parameters on the URL, rather than as file system specifier. The format becomes:

10 <protocol><server>[<routing>]<card and button
indicators>?<parameters>

where "?" is the standard indicators of the start of a list of parameters. In our example above where the user is male, the request becomes:

<http://www.planetportal.com/cards/12/1.html?gender=male>

15 or a similar format. Parameters can be combined into URLs such as:

<http://www.planetportal.com/cards/12/1.html?gender=male&zipcode=27514&phonenu,ner=9195551212>

The web server would use the file system to select the file as described above. In our example, it would find the file "1.html" in the
20 subdirectory "cards/12". However, in this case, the file could access the URL's parameters.

In most cases, HTML files don't access parameters, but server-side scripts such as CGI programs do. Thus, to generate demographic-based responses, the HTML file is typically replaced by a server-side script. In our

25 example, this might look like:

<http://www.planetportal.com/cards/12/1.cgi?gender=male>

The CGI program then uses standard web techniques to parse the parameters, and select among multiple redirect sites. Such a script might look like:

```

5  #!/usr/bin/perl

    require "cgi-lib.pl";          # standard cgi library

    &ReadParse(\%input);            # Read parameters into name/value
10 pairs in                          # the %input hash table

    # In this basic case, we assume that the script is looking to send
    the user
15 # to different sites based upon gender.

    if ($input{'gender'} eq 'male') {
        $URL = 'http://www.men.com';
    } elsif ($input{'gender'} eq 'female') {
20     $URL = 'http://www.women.com';
    } else {
        $URL = 'http://www.alien.com';
    }

25 print "Location: $URL\n\n";

```

This embodiment greatly reduces complexity in the file system, but at the expense of running server-side scripts. Such scripts consume server cycles, and thus increases costs.

30 Here we can extend the notion of intelligent client direction of card
URLs to discrete servers. In this case, we would use a separate chain of
servers to handle CGIs, which would alleviate load on the standard content
servers. A key assumption here is that the amount of static content will
greatly outnumber CGI generated dynamic content, thus ensuring that the
35 differences in overhead of delivering dynamic vs. static content would be
offset by the significantly smaller number of requests for dynamic content.
The more servers that need to be dedicated to dynamic content, clearly, the
higher the cost.

Yet another alternative, illustrated in figure 5, is to replace the demographic parameters with a user identification (user ID). In this case, when the user enters his demographic information, he is assigned a unique user ID. The user ID and demographic information are stored in a database
5 attached to the web server.

On subsequent requests, the user ID is attached to the standard flow. Such a URL might look like:

`http://www.planetportal.com/cards/12/1.cgi?userID=dlk12345`

Upon receiving such a flow, the server-side script parses the user ID from
10 the URL, and looks up the user's demographic record in the database. The demographic information is then used to select among multiple redirect sites. The selected redirect site is returned to the client, as described above. The advantage of this embodiment is that the entire demographic record is available to the CGI program, even if the URL only specified a single
15 parameter (the user ID). However, this embodiment requires the deployment of a demographic database, which can increase costs.

Note that the two previously techniques can be combined seamlessly. If the CGI program receives a user ID, the demographic information is retrieved from the data; if only demographic information is received, the CGI
20 program uses that to make its selection; and if nothing is received, the default redirect indicator is returned.

Data Mining

The techniques above select among static web-page alternatives. These techniques can be supplemented by employing data mining techniques.

5 Data mining is a generic term that indicates data are scanned for patterns. For example, through data mining of its receipts, a grocery store might learn that bagels and cream cheese are frequently purchased together. After learning this fact, the store might locate these items closer together.

Techniques for data mining are well-known in the industry, and will not be discussed in depth here. This novel feature of this aspect of this invention is application of data mining to web page selection.

Each card issuer, or an agent acting for the issuer, executes the following steps:

- 1) deploy a card which each button mapped to an initial URL
- 2) collect data about card use
- 3) partition the data
- 4) perform pattern inferences
- 5) dynamically adjust URL mappings according to inferred data

20 We discuss each step below.

First, the card issuer prints a set of cards, each containing the same code. The issuer, or an agent of the issuer, distributes the cards to some set of potential customers.

Second, as the customers swipe the cards, data are collected a
25 central site. When a user swipes a card, an HTTP request flows from the

client to the server, as is described in detail above. In the preferred embodiment, an Apache web server is used. As with many web servers, Apache can be configured to log all incoming requests. Preferably, this feature is enabled.

5 With logging enabled, a record of each transaction is written to stable media, such as a disk drive. Recall from above, the HTTP requests created by the Request Generator contain, without limitation, the card number, button number, user ID, and other information. Thus, by enabling logging on the Apache server, and exploiting the careful creation of the HTTP requests,
10 this invention enables the collection of data without any changes to a standard web server. Thus, we accomplish one object of this invention.

 Third, once the data are collected, they are partitioned. Partitioning data can be as simple as retrieving all log records for a single card, and this will be the typical case. However, in cases where an issuer deploys multiple
15 cards (e.g., Coke and Diet Coke cards), the issuer can be allowed access to complex sets of data.

 Similarly, the firm collecting the data can allow access to results from other card issuances, either individually, or in aggregate. Such data can help an issuer tailor their promotions. For example, if one cola manufacturer
20 gain access to the results of card promotion by beer manufacturers, the cola manufacturer can learn about what creates swipes effectively. In a simple case, the cola manufacturer might learn that 10% discounts increase swipe rates 50%, and that 20% discounts increase rates 51%. From that information, the cola manufacture might conclude that a 10% discount is
25 sufficient.

Fourth, the issuer, or an agent of the issuer performs pattern inferences. One example of such an inference is the discount example given above. Such inferences are well-known in data mining art, and will not be discussed in detail here.

Fifth, the inferences are used to modify the URLs mapped by the card and button numbers. Modifying the URL mapping, in the preferred embodiment, simply entails changing the redirect site in the HTML or CGI file stored in the file system, as described in great detail above. The new URLs are chosen to better entice one or more demographic categories of user to swipe the card.

In one example, consider a beer card with two button mappings. On the first swipe, all users are taken to the beer company's home page; the first button takes a user to a 10% discount on the next purchase; and the second button takes users to offer for a free calendar containing risqué pictures. Upon mining the data, the beer company might learn that females predominately press button 1, and men predominately press button 2. In

response, the beer company can change the initial-swipe site for females to be the discount page, and the page for men to be the calendar offer.

Virtually any data mining technique can be used, with associated dynamic adjustment to the URL mappings. This example does not limit the

5 invention.

Scoring

Another technique novel feature of this invention is the use of "scoring." In the preferred embodiment, the score is derived from a regression model, although other models can be applied without materially

10 effecting this invention.

In our use of scoring, each user is observed to have both demographic characteristics (e.g., gender), and observed behaviors (e.g., how many times he swiped card 10). Card issuers are permitted to supply a model by which users are scored, and to supply URLs appropriate for each

15 score. The use of scoring is illustrated in figure 6.

In the preferred embodiment, when a user swipes a card, the ensuing request carries the user's unique ID. That ID is used to query a database containing the user's demographics and observed behaviors. The information relevant for the card is retrieved, and used to calculate the score according to the supplied model. Based on this calculation, the URL appropriate for this score is returned to the user's browser using the redirect mechanism described above.

For clarity, and without limitation, we provide an example. Imagine a user named Mary who has a known set of demographics and behaviors. In

25 this example, her income is \$100,000; has 6 kids; works as a stock-trader;

"00400400" 00400400

visited a given site 37 times; visited another site 15 times. First, the model must translate her occupation, or her class of occupation (e.g., white collar) into a score. In this example, we assume that a stock-trader is "worth" 21 points.

5 We then create a regression model of this form:

$$\text{Score} = b_1 \cdot v_1 + b_2 \cdot v_2 + \dots + b_N \cdot v_N$$

In this case, we have five variables, which reduces the formula to:

$$\text{Score} = b_1 \cdot v_1 + b_2 \cdot v_2 + b_3 \cdot v_3 + b_4 \cdot v_4 + b_5 \cdot v_5$$

Next, each parameter must be assigned a value appropriate for this
10 model. In this example, we assigned the weighting to be: .006, .03, 7.65, .25, and .61. These weighting are preferably determined by the card issuer in a way that reflects the relevance of each item parameter.

Substituting, we get:

$$\text{Score} = .006 \cdot 100,000 + .03 \cdot 6 + 7.65 \cdot 21 + .25 \cdot 37 + .61 \cdot 15$$

15 Which rounds to a score of 779.

Once the score is computed, it is used to select the appropriate URL. Preferably, the selection is accomplished by a CGI program running on the web server, but other equivalent technologies (such as servlets) can be used. Such as CGI program might look like:

```
20  #!/usr/bin/perl

    require "cgi-lib.pl";          # standard cgi library

    &ReadParse(\%values);          # Read parameters into name/value
25  pairs in                        # the %input hash table

    # In this basic case, we assume that the script is looking to send
    the user
30  # to different sites based upon scoring of multiple parameters, here
    assigned
    # the values b1 ... b5. Values for corresponding weighting of
    behavior params
```

are kept in the hash table %weight, which may be imported many ways, but in
this case is simply defined by assignment.

```
5  %weight = (  
    'b1' => .006,  
    'b2' => .03,  
    'b3' => 7.65,  
    'b4' => .25,  
10  'b5' => .61  
    );  
  
    $score = 0;      # initialize score  
  
15  foreach $key (keys %values) {      # Iterate through all  
    # params passed to script  
    if ($key =~ /b\n+/) {             # If this is a behavior  
value b0..bn  
        $score += ($values{'$key'} * $weight{'$key'});  
20    }                                # Add computed value to  
    the score  
    }  
  
    if ($score > 700) {  
25        $URL = 'http://www.company.com/ideal_customer.html';  
    } elsif ($score > 300) {  
        $URL = 'http://www.company.com/good_customer.html';  
    } else {  
30        $URL = 'http://www.company.com/welcome.html';  
    }  
  
    print "Location: $URL\n\n";
```

In this example, Mary's score is considered high, so she is redirected
35 to a site offering premier service, such as a hefty discount for a first
purchase. So, in this example, by employing the scoring technique, the card
issuer can restrict its best offers for users the model predicts to become
better customers, thus providing the most revenue to the card issuer. This
illustrates the value of scoring in response to card requests.

40 Since the card issuer is provided more value, the firm coordinating the
card requests gains another opportunity to generate additional revenue from
the card issuer. In one model, card issuers are charged a fee for employing
scoring to select URLs. In another model, one fee (possible zero) is charged
for employing a standard scoring technique, and a second fee, typically

higher will be charged for the issuer to supply their own model. In yet another model, the card issuer is charged by the number of different URLs they choose to deploy. (Such a model can be combined with others outlined above.) Similar business methods will be obvious to one skilled in the art.

5

An Automated Card Enablement Process

Once sufficient processes have been deployed to handle a large volume of card requests, the bottleneck becomes the efficient process of producing and distributing cards. This aspect of the invention provides an automated process for creating cards, manufacturing them, and deploying the associated URLs.

Figure 7 shows the process of card fulfillment in outline. The main components of the process are the:

- Request coordinator
- 15 • Art generator
- Identification database
- Printer
- Fulfillment coordinator
- Card handling web server

20

The request coordinator orchestrates the entire process, interacting with the other components of the system.

Note that in many cases, a business will operate the request generator, and other components of this system. We refer to this party as the "firm."

This firm can generate revenue from business model associate with the process.

It executes the following steps:

- 1) Gather issuer input
- 5 2) Obtain a unique identifier
- 3) Register the card's URLs with the proper web server

Optionally, the request coordinator can execute one of more of the follow steps to simplify fulfillment of the card order.

- 4) create an image for the card
- 10 5) route the order to a printer
- 6) arrange fulfillment of the cards

We discuss each step below.

The first set of steps allows a card issuer to obtain a code, and prepares the server(s) to handle requests associated with the code.

- 15 The first step is to gather information from the card issuer. At a minimum, this includes one or more URLs associated with the card, and an indication of how those URLs are to map to buttons.

- Optionally, the request coordinator can gather other data, such as information about the issuer, billing information for the account, a user ID for
20 this account and an associated password. This information is stored in the customer database.

Next, the request coordinator queries the ID database to obtain a unique code. Preferably, to avoid conflicts, the ID database issues codes sequentially.

As an improvement to the simple scenario, a card issuer can request blocks of similar code, which are then reserved by ID database. As illustrated in figure 8, to use the pre-reserved code, the card issuer supplies the code, along with identification (preferably a user ID and password). The request coordinator validates the user's identity using the customer database, then proceeds with the process described below.

Optionally, the firm can charge a fee for code issuance.

Finally, the request coordinator establishes the proper file infrastructure to handle card request. Preferably, this entails storing files in directories contain redirect indicators in the structure described at length above. For example, if card 15 was issued, and the card issuer indicated the www.planetportal.com was to be associated with button 1, then a file containing a redirect indicator (as described above) would be created in a directory called "15", with the file named "1.html". The web server is then prepared to handle requests for this card.

Note that the web server method described above does not limit the invention. The code mappings can be stored in a database that is accessed by a web server without otherwise materially effecting this invention.

The firm can charge a fee for establishing and/or hosting the URL mappings either on a web server's file system or in a database.

Optionally, the request coordinator can handle other portions of the card fulfillment process, requiring slight modifications to the process described above.

First, each card must carry a printed version of the code. So, in addition to the artwork typically carried on card, the code must be printed.

First, the request generator receives from the issuer an image format for the card. Examples of image formats include, but are not limited to GIF, BMP, and JPEG. This supplements the input gathering step listed above.

Next, the code in step 2 is translated into a binary code. Preferably, the integer code is converted into its binary representation. For example, in a 3-bit code, 0 is represented as 000, 1 as 001, 2 as 010, etc. The binary code is then converted into a machine-readable format. Many technologies exist for such conversion. In one simple example, the binary digit '0' is represented by a dark area of the card, and a '1' is represented as a light area. A reader for such a light/dark encoding is described in our co-pending application. This machine-readable format is then converted into an image representation, which is superimposed on the image supplied by the issuer. This process is illustrated in figure 9. This completes step 4.

Next, should the issuer chose, that artwork can be routed to a printer. This requires modification to step 1 such that the issuer optionally provides the quantity of cards desired, along with billing information. The request coordinator stores the quantity locally, and stores the billing information in the customer database. The image file for the card is then routed to an appropriate printed.

We present two alternatives for billing. First, the printer is given the issuer's billing information, and bills the issuer directly. Optionally, the firm operating the request coordinator can charge a fee to the printer for the referral. Second, the printer can bill said firm, which in turn bills the issuer. Optionally, said firm can include a markup to generate revenue from the act of printing.

5

10

20

It will be understood that various details of the invention may be
5 changed without departing from the scope of the invention. Furthermore, the
foregoing description is for the purpose of illustration only, and not for the
purpose of limitation—the invention being defined by the claims.

CLAIMS

What is claimed is:

1. A system for facilitating access to a web site, the system comprising:
 - 5 (a) a card reader for reading a card type code from a card insertable in the card reader; and
 - (b) a request generator operatively associated with the card reader for receiving the card type code and for formulating a query containing location information usable by a web server for
10 locating an address file containing at least one web site address corresponding to the card type code without using a database.
2. The system of claim 1 wherein the location information generated by
15 the request generator includes a location and file name in a web server file system.
3. The system of claim 2 wherein the location information generated by the request generator includes a URL containing the location and file name
20 in the web server file system.
4. The system of claim 1 comprising a keypad associated with the card reader for generating a key code in response to user access to the keypad.

5 6. The system of claim 5 wherein the location information includes a location and a file name in a web server file system, the location corresponding to the card type code and the file name corresponding to the key code.

8. The system of claim 1 comprising:

(d) a web server for receiving the query and sending the address file to the user application in response to the query.

9. The system of claim 8 wherein the server is adapted to send extensible markup language (xml) data to the user application in response to the query and the user application is adapted to display the xml data to the card user.

5 11. The system of claim 10 wherein the request generator generates the
web server selection code based on the card type code.

13. The system of claim 10 comprising at least one load balancer for receiving the query and selecting a web server for processing the query.

15. The system of claim 1 wherein the request generator is adapted to
20 include at least one parameter for invoking a cgi program on the web server.

16. The system of claim 15 wherein the cgi program is adapted to select a URL based on demographic information relating to a slide card user.

5

- 10

15

20

- (a) receiving a query containing a card type code and demographic indicia for obtaining demographic information regarding a card user;
- (b) accessing a database to obtain the demographic information based on the indicia in the slide card code;
- (c) determining a score for the card user based on the demographic information; and
- (d) returning a web page to the card user based on the score.

21. The method of claim 19 wherein determining a score for the user includes performing a lookup in a score database based on the demographic information to obtain the score.

23. The method of claim 19 wherein the demographic information includes at least one of: the card user's occupation, the card user's income, the card user's class of occupation, and information regarding the card user's family size.

20

25. The method of claim 24 wherein the behavioral information includes the number of time the user has accessed one or more web sites.

5

- 10

15

29. The method of claim 26 storing web site address information includes storing at least one URL associated with the issuer in the web server file system.

25

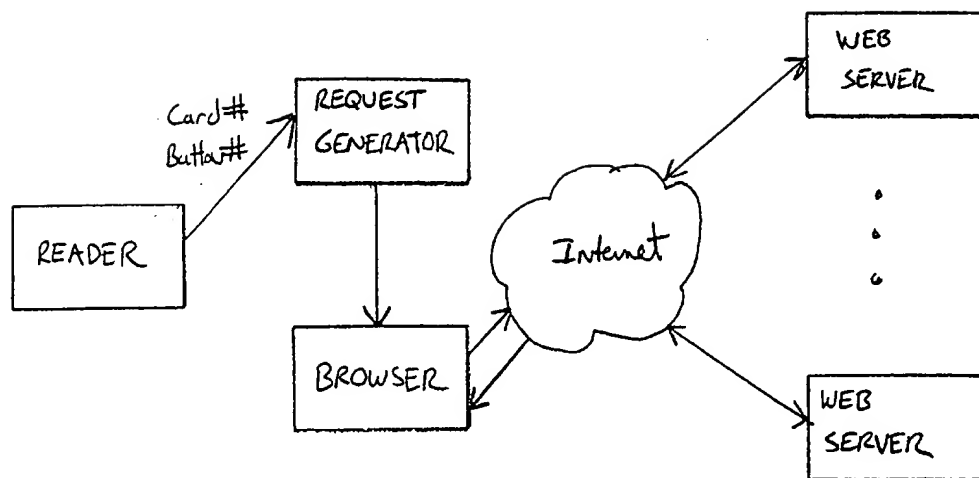


FIG. 1

60430430.000400

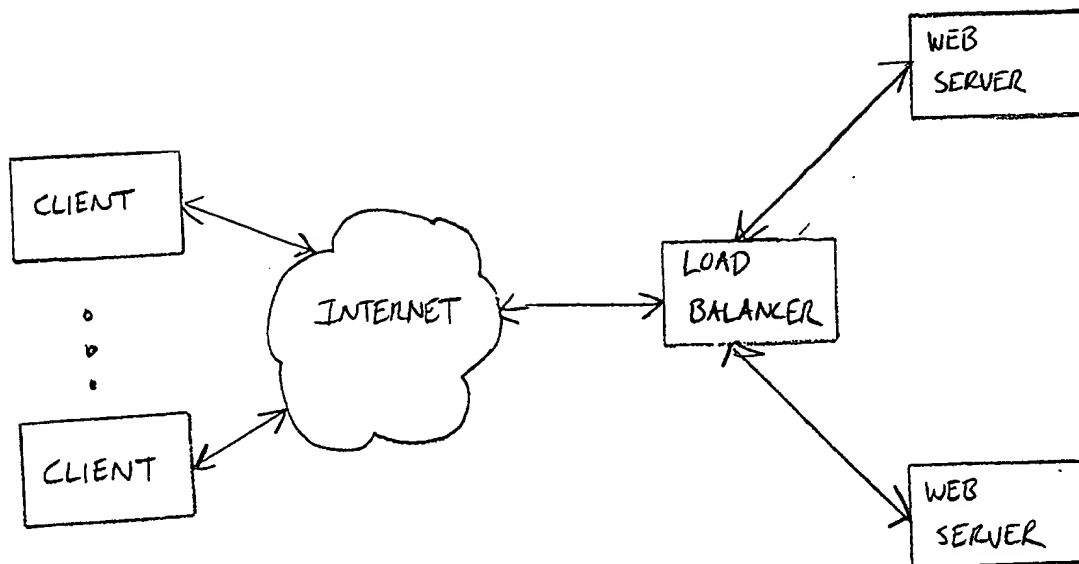


FIG. 2

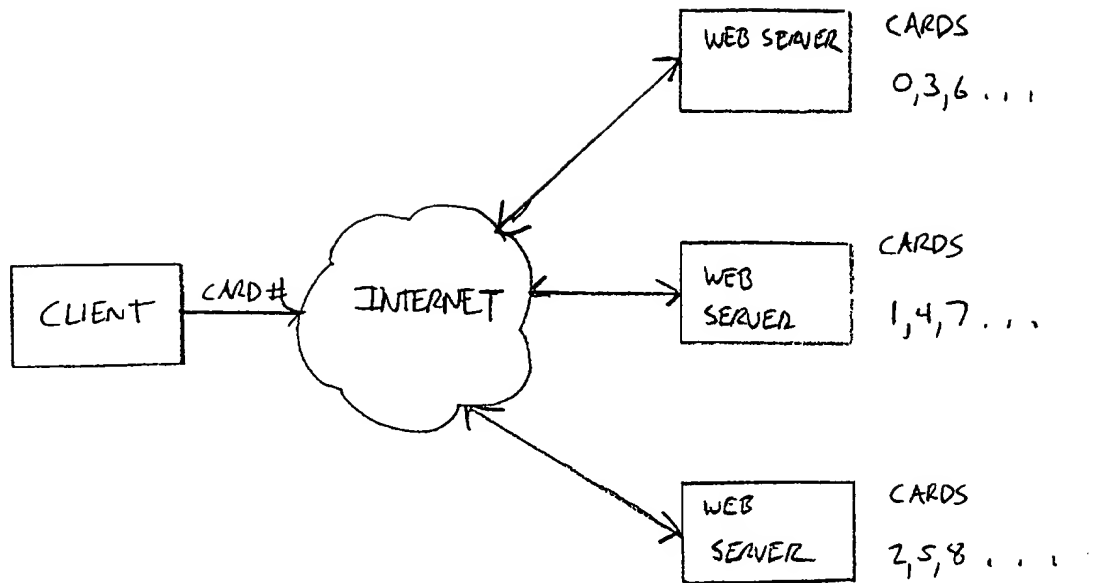


FIG. 3

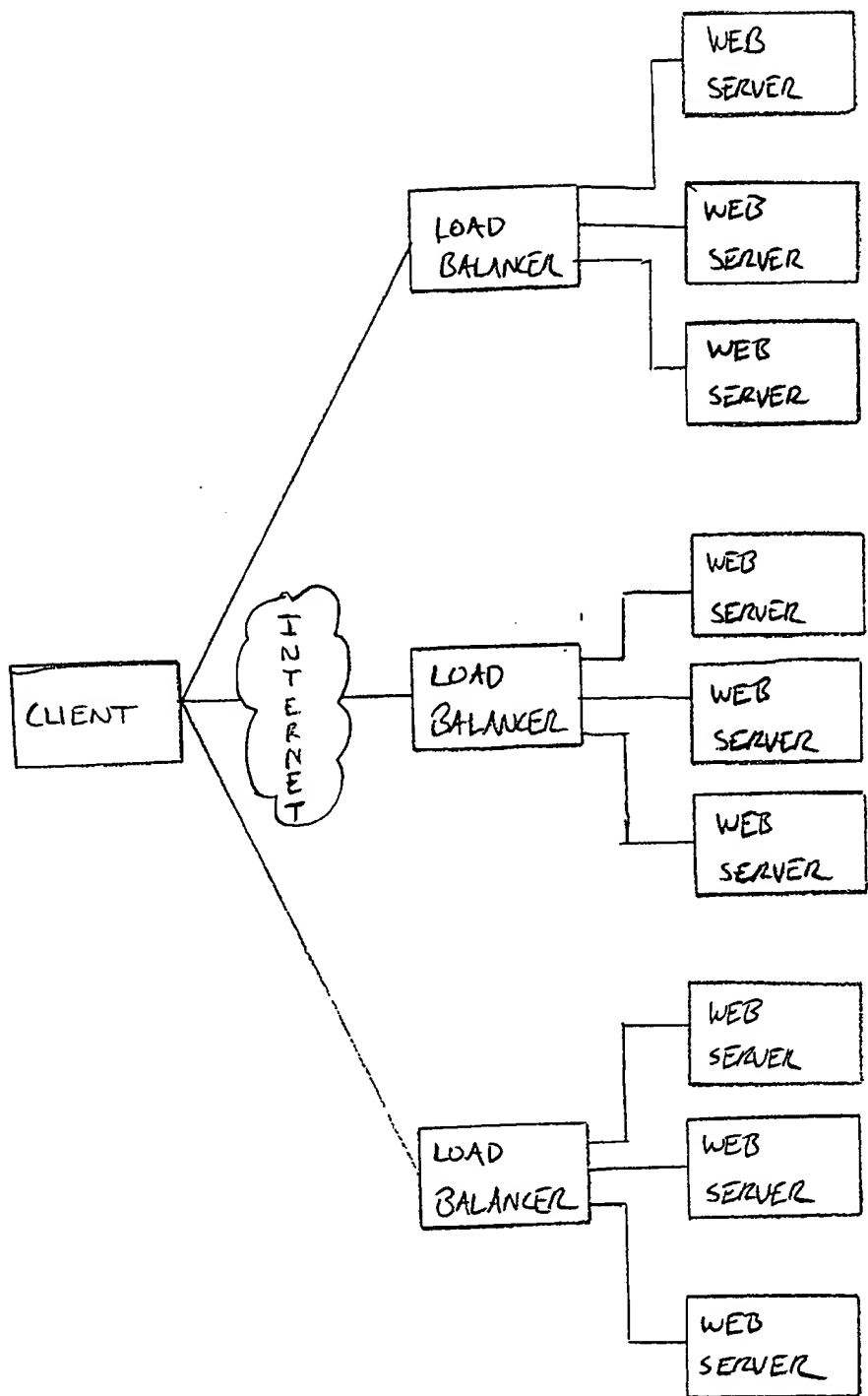


FIG.4

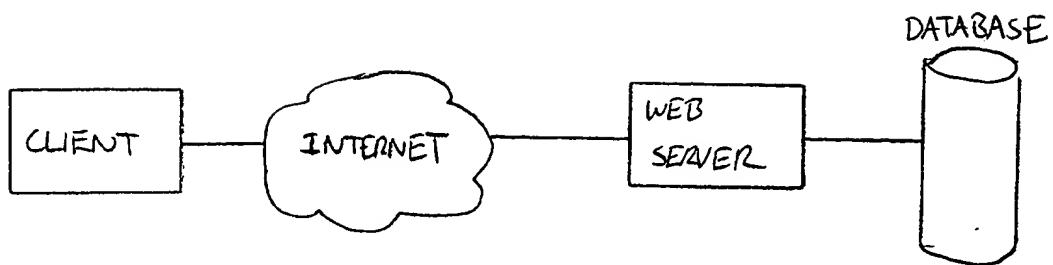


FIG. 5

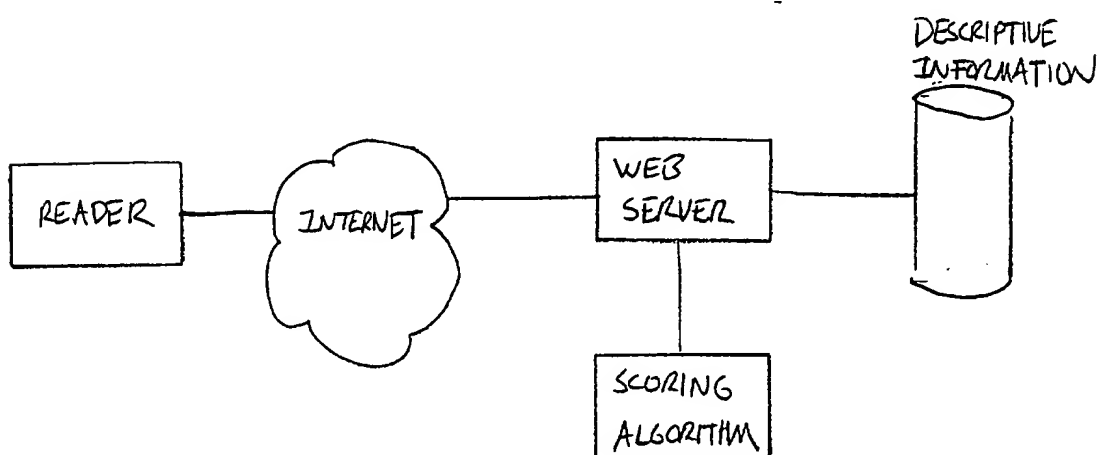


FIG. 6

004030 904030 004030

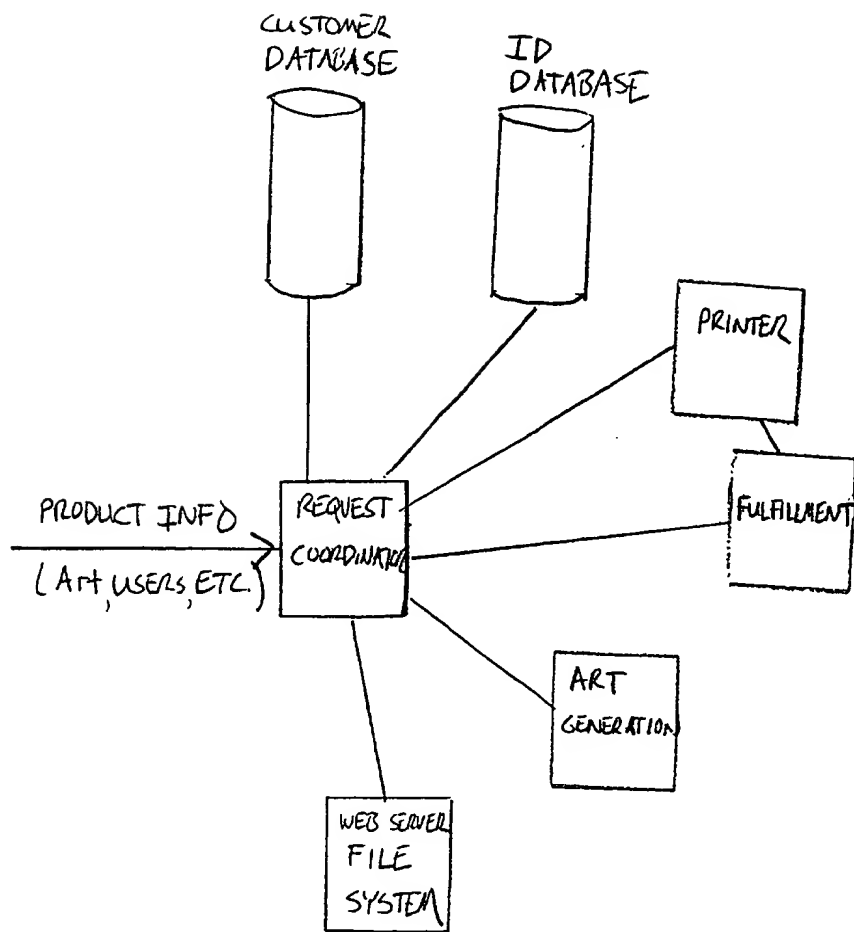


FIG. 7

```
graph TD; Input[1: CARD ID, USER ID, PASSWORD, URLS] --> RC[REQUEST COORDINATOR]; RC -- 2 --> ID[(ID DATABASE)]; ID -- 3 --> RC; RC -- 4 --> WS[WEB SERVER FILE SYSTEM];
```

F16.8

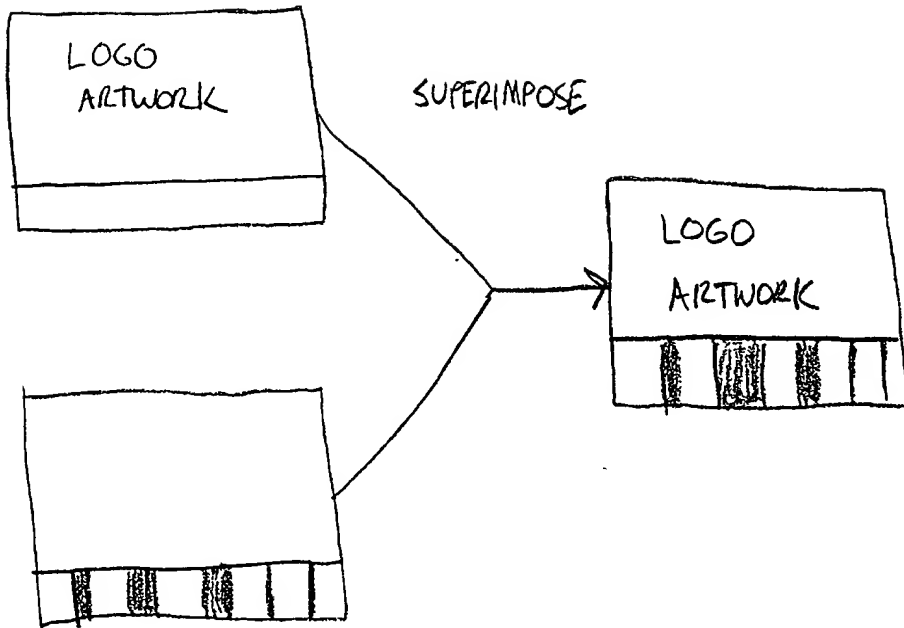


FIG. 9